PB84-171743

Air Land Water Analysis System (ALEAS)
A Multi-Media Model for Toxic Substances

Little (Arthur D.), Inc., Cambridge, MA

Prepared for

Environmental Research Lab., Athens, GA

Mar 84

U.S. Department of Commerce
National Technical Information Service
NTIS

PB84-171743

EPA-600/3-84-052
March 1984

AIR LAND WATER ANALYSIS SYSTEM (ALWAS): A MULTI-MEDIA MODEL FOR TOXIC SUBSTANCES

by

William A. Tucker
Alan Q. Eschenroeder
Gary C. Magil
Arthur D. Little, Inc.
Acorn Park
Cambridge MA 02140

Contract No. 68-03-2898

Project Officer

Kenneth F. Hedden
Technology Development and Applications Branch
Environmental Research Laboratory
Athens GA 30613

ENVIRONMENTAL RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
ATHENS, GEORGIA 30613

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

TECHNICAL REPORT DATA

(Please read Instructions on the reverse before completing)

1. REPORT NO.
EPA-600/3-84-052

2.

3. RECIPIENT'S ACCESSION NO.
PB84 171743

4. TITLE AND SUBTITLE
Air Land Water Analysis System (ALWAS): A Multi-media Model for Toxic Substances

5. REPORT DATE
March 1984

6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S)
William A. Tucker, Alan Q. Eschenroeder and Gary C. Magil

8. PERFORMING ORGANIZATION REPORT NO.

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Arthur D. Little, Inc.
Acorn Park
Cambridge MA 02140

10. PROGRAM ELEMENT NO.
ACUL1A

11. CONTRACT/GRANT NO.
68-03-2898

12. SPONSORING AGENCY NAME AND ADDRESS
U.S. Environmental Protection Agency--Athens GA
Office of Research and Development
Environmental Research Laboratory
Athens GA 30613

13. TYPE OF REPORT AND PERIOD COVERED
Final, 7/80-8/83

14. SPONSORING AGENCY CODE
EPA/600/01

15. SUPPLEMENTARY NOTES

10,000 sq km

16. ABSTRACT

The Air Land Water Analysis System (ALWAS) is a multi-media environmental model for describing the atmospheric dispersion of toxicants, the surface runoff of deposited toxicants, and thesubsequent fate of these materials in surface water bodies. ALWAS dipicts the spatial and temporal distribution of contaminant concentrations in a watershed and the air above it. Linked in ALWAS are three submodels that provide for independent and partially coupled usage modes--Dispersion and Deposition of Toxics (DiDOT); a modification, called NPSDEP, of the Nonpoint Source Model; and Exposure Analysis Modeling System (EXAMS). ALWAS is appropriate for evaluating multi-media water quality problems for watersheds as large as 10^4 km^2. The full range of ALWAS capabilities may be exercised for hydrophobic organic chemicals of relatively low vapor pressure. By careful selection and linkage of the submodels, ALWAS may be applied to investigate the environmental behavior of a broad range of toxicants including highly soluble or volatile organics and heavy metals. The report incorporates detailed software documentation and a users manual. The sensitivity of model results to uncertainties in key input parameters is investigated for a hypothetical simulation of benzo(a)pyrene behavior in a small watershed.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group

18. DISTRIBUTION STATEMENT
RELEASE TO PUBLIC

19. SECURITY CLASS *(This Report)*
UNCLASSIFIED

20. SECURITY CLASS *(This page)*
UNCLASSIFIED

21. NO. OF PAGES
460

22. PRICE

EPA Form 2220-1 (9-73)

DISCLAIMER

The information in this document has been funded wholly or in part by the United States Environmental Protection Agency under Contract No. 68-03-2898 to Arthur D. Little, Inc. It has been subject to the Agency's peer and administrative review, and it has been approved for publication as an EPA document. Mention of trade names or commercial products does not constitute endorsement or recommendation for use by the U.S. Environmental Protection Agency.

FOREWORD

Environmental protection efforts are increasingly directed towards prevention of adverse health and ecological effects associated with specific compounds of natural or human origin. As part of this Laboratory's research on the occurrence, movement, transformation, impact, and control of environmental contaminants, the Technology Development and Applications Branch develops and evaluates management or engineering tools for assessing and controlling adverse environmental effects of toxic substances.

Solutions to serious environmental pollution problems increasingly involve analyzing and evaluating difficult trade-offs between useful substances and products and risks posed to human health and environmental quality from toxic substances. One useful method for estimating the actual human and environmental exposure to substances and their effects is the application of mathematical models. This report develops and evaluates a multimedia mathematical exposure assessment model which may be used to describe the atmospheric dispersion and deposition of toxicants, surface runoff of deposited contaminants, and the subsequent fate of toxicants in surface water bodies.

William T. Donaldson
Acting Director
Environmental Research Laboratory
Athens, Georgia

ABSTRACT

The Air Land Water Analysis System (ALWAS) is a multi-media environmental model for describing the atmospheric dispersion of toxicants, the surface runoff of deposited toxicants, and the subsequent fate of these materials in surface water bodies. ALWAS depicts the spatial and temporal distribution of contaminant concentrations in a watershed and the air above it. Linked in ALWAS are three submodels that provide for independent and partially coupled usage modes--Dispersion and Deposition of Toxics (DiDOT); a modification, called NPSDEP, of the Nonpoint Source Model; and Exposure Analysis Modeling System (EXAMS).

DiDOT quantifies both wet and dry deposition rates over hourly, daily, and longer averaging periods. NPSDEP describes surface runoff of contaminants and EXAMS simulates their fate in surface waters.

DiDOT, which was developed in the study, is based on a Gaussian plume approach modified to account for deposition. Its meteorological and emission input requirements are similar to those of standard, short-term air quality assessment models. Input parameters include settling velocity, dry deposition velocity, and precipitation scavenging ratio. Guidance for estimating these parameters is provided.

ALWAS is appropriate for evaluating multi-media water quality problems for watersheds as large as $10^{14}km^2$. The full range of ALWAS capabilities may be exercised for hydrophobic organic chemicals of relatively low vapor pressure. By careful selection and linkage of submodels, ALWAS may be applied to investigate the environmental behavior of a broad range of toxicants including highly soluble or volatile organics and heavy metals.

The report incorporates detailed software documentation and a users manual. The sensitivity of the model results to uncertainties in key input parameters is investigated for a hypothetical simulation of benzo(a)pyrene behavior in a small urban watershed.

This report was submitted in fulfillment of Contract No. 68-03-2898 by Arthur D. Little, Inc., under the sponsorship of the U.S. Environmental Protection Afency. This report covers the period from July 1980 to August 1983, and work was completed as of August 1983.

CONTENTS

Page

Foreword iii
Abstract iv
Figures vii
Tables viii

1.0 INTRODUCTION 1

- 1.1 Background 1
- 1.2 Purpose and Scope 1
- 1.3 Findings 1
- 1.4 Recommendations 2

2.0 OVERVIEW OF AIR LAND WATER ANALYSIS SYSTEM (ALWAS) 5

- 2.1 Purpose 5
- 2.2 Scope and Limitations 5
- 2.3 Features 9
  - 2.3.1 DiDOT 9
  - 2.3.2 NPSDEP 11
  - 2.3.3 EXAMS 13
- 2.4 Input/Output 13
  - 2.4.1 Inputs 13
  - 2.4.2 Outputs 16

3.0 TECHNICAL BASIS OF ALWAS 17

- 3.1 Model for Air Dispersion and Deposition of Toxics (DiDOT) . 17
  - 3.1.1 Point Source Model withour Deposition 18
  - 3.1.2 Treatment of Effective Stack Height 18
  - 3.1.3 Point Source Model Incorporating Dry Deposition Processes 22
  - 3.1.4 Area Source Model 29
  - 3.1.5 Chemical Decay Term 33
  - 3.1.6 Wet Deposition 35
- 3.2 NPSDEP 36
- 3.3 Dynamic Version of EXAMS 37
- 3.4 Interfacing NPSDEP and EXAMS 37

4.0 USERS GUIDE FOR ALWAS 40

- 4.1 Getting Started 40
  - 4.1.1 Basic Definitions and Concepts 40
  - 4.1.2 Data Sources and Preparation 42
  - 4.1.3 Systems Description 49

Page

4.2 Program Description 53
    4.2.1 DiDOT 53
    4.2.2 NPSDEP 61
    4.2.3 EXAMS 64
    4.2.4 NPSEXAMS 64
4.3 Input Data Formats 68
    4.3.1 DiDOT 68
    4.3.2 NPSDEP 78
    4.3.3 EXAMS 84
    4.3.4 NPSEXAMS 84
4.4 Guidance in Estimating Input Parameters 85
    4.4.1 Deposition Velocity 86
    4.4.2 Settling Velocity 95
    4.4.3 Scavenging Ratio 95
    4.4.4 Surface Roughness Length 96
    4.4.5 EXAMS Load Averaging Time 96
    4.4.6 Fugitive Dust Emission Ratios 96
4.5 User Operating Instructions 99
    4.5.1 DiDOT 99
    4.5.2 NPSDEP 118
    4.5.3 NPS/EXAMS 119

5.0 SENSITIVITY ANALYSIS 122
  5.1 Preparation of Inputs for Sensitivity Analysis 123
    5.1.1 DiDOT 123

REFERENCES 141

APPENDICES
  A. Review and Evaluation of Models for Atmospheric Dispersion, Degradation and Deposition of Toxic Pollutants 145

  B. Size Distribution of Various Aerosols 182

  C. ALWAS Program Listing 187

FIGURES

Number Page

2-1 Environmental System Simulated by ALWAS 9
2-2 Comparison of Vertical Concentration Profiles in Alternative Models 12
2-3 NPSDEP 14
3-1 Variation of Roughness Length with Terrain Features 31
3-2 Typical NPS Loading Data 38
3-3 Typical EXAMS Response 38
4-1 Relationship between Land Area Definitions Used in ALWAS . . 41
4-2 ALWAS Flow Chart 50
4-3 DiDOT Flow Chart 59
4-4 EXAMS Dynamic Version 65
4-5 Input Data for DiDOT 83
4-6 Dependence of Vapor/Aerosol Partitioning on Contaminant Vapor Pressure and Available Aerosol Surface Area 87
4-7 Correspondence between Estimated and Observed Gaseous Deposition Velocities 91
4-8 Approximate Climatologic Average Dry Deposition Velocity as a Function of Aerosol Diameter 93
4-9 Variations of Scavenging Ratio with Particle Size 97
4-10a DiDOT Overlays 100
4-10b DiDOT Memory Diagram 101
4-11 DiDOT Overlay Description Language (ODL) 102
4-12 Example Outputs from DiDOT 106
4-13 Overlay Description of EXAMS 120
4-14 Overlay Structure of EXAMS 121
5-1 Third Fork Creek Basin 126
5-2 Area Source Grid 129
5-3 Area Source Emission Inventory 134
5-4 Area Source Grid in Relation to Study Area 135
5-5 Receptor Grid Overlay 137
5-6 BaP Concentration in Hypothetical Pond 138
5-7 Results of Sensitivity Analysis 140

TABLES

Number Page

3-1 ASME Coefficients and Exponents Used to Calculate Horizon and Vertical Plume Dispersion Parameters 19
3-2 Selection of Stability Categories 24
3-3 Mean Values and Standard Errors of the Reciprocal Stanton Number B 30
4-1 Preprocessor Initialization Card Format 69
4-2 Preprocessor Mixing Height Data Card Format 69
4-3 Preprocessor Random Digits File Format 70
4-4 Input Formats for $CNTROL Package 73
4-5 Input Data Formats for $SPTSRCE Package 75
4-6 Input Data Formats for $ARSRCE Package 76
4-7 Input Data Formats for $CITY Input Package 77
4-8 Input Data Format for $METCTL Input Package 79
4-9 Input Formats for $CATCHS Control Package 81
4-10 Particle Size Ranges for Open Dust Sources 98
4-11 Reference Rules for DiDOT Input Seck Set-Up 104
5-1 Input Parameter Values for Sensitivity Testing 124
5-2 Point Source Characteristics 128
5-3 Emissions from Transportation 130
5-4 Emissions form Miscellaneous Sources 131
5-5 Summary of Emissions Used as Input to DiDOT 133

1.0 INTRODUCTION

1.1 BACKGROUND

Notable progress has been made recently in the field of environmental modeling. Most of it has been made in the modeling of a single medium (i.e., air quality, surface water quality, groundwater quality, or biological systems); however, many pollutants are released to more than one medium and are subject to intermedia transfer. Significant effects may occur in more than one medium or may be most important in a medium other than the one to which they were released. These multi-media environmental problems are receiving increasing attention as more of them are recognized (e.g., acid rain, PCB's in the Great Lakes, volatilization and groundwater recharge from hazardous waste lagoons and pesticide losses by volatilization). Multi-media environmental models for assessing such problems have not been available.

1.2 PURPOSE AND SCOPE

This project was intended to partially fill the gap by developing a model which describes the effect of atmospheric emissions, non-point sources (runoff), and point discharges of toxic pollutants on surface water quality.

The scope of work for the project reported herein included a review and evaluation of existing air quality models to identify any which were suitable for linking with NPS (non-point source model) and EXAMS (Exposure Assessment Modeling System). These two models have been developed under the aegis of EPA's Athens Environmental Research Laboratory. A suitable air model was one that describes both wet and dry deposition processes for toxic pollutants and is applicable at a river basin scale (length scales of up to 100 km) over short-time frames in order to interface with NPS. Once an appropriate air model was identified, it was to be linked with NPS and EXAMS to achieve a multi-media model describing the effect of multi-media toxicant releases, particularly air emissions, on surface water quality. The resultant model was to be subjected to sensitivity analysis and documented.

1.3 FINDINGS

The review of available air models revealed that there were no models which fit these criteria and which were "available", i.e., accessible and documented. Most air quality models do not account for deposition processes. Some which do treat deposition are specifically designed for the acid rain problem and are neither pertinent to toxic pollutants nor to the river basin spatial scale. Models designed to simulate acid rain must follow pollutants up to $\sim$1000 km. Consequently, they use multiple station weather data to define plume trajectories. A local weather station is likely to adequately represent the winds and mixing conditions within a river basin unless the terrain is mountainous. The average separation distance of National Weather Service stations is greater than the model application scale so model domain will rarely include more than one surface weather station.

The project resulted in the model referred to as the Air, Land, Water Analysis System (ALWAS). ALWAS is one of the first models of its type, and has not been calibrated or verified for any field situation. For these reasons, ALWAS is considered a "first generation" multi-media model with room for improvement and extension.

From this review of air quality models, it was apparent that one must be developed specifically to link with NPS and EXAMS -- the result was DiDOT, the model for Dispersion and Deposition of Toxics. The review also showed that minor modifications of NPS and EXAMS were required in order to facilitate their linkage into ALWAS.

ALWAS can simulate the effects on surface water quality of multi-media toxicant releases to the environment. It is most suitable for persistent organic chemicals which tend to adsorb to particulate matter, but ALWAS or various combinations of its sub-models, may also provide valuable multi-media information for metals and more soluble organics, given care in its application.

An overview of ALWAS, including its purpose, scope, limitations, and features is presented in Section 2. A technical discussion of the assumptions and technical rationale for the software developed under the current contract and presented here for the first time is found in Section 3. The User's Guide for ALWAS is presented in Section 4 and emphasizes portions of the software developed specifically for ALWAS. Most of NPSDEP (the modified version of NPS used in ALWAS) and the dynamic version of EXAMS are already described in existing EPA publications, and the user is referred to those publications for further details on NPS and EXAMS. The User's Guide incorporates methods for estimating some of the critical ALWAS input parameters such as the deposition velocity, settling velocity, scavenging ratio, and air degradation half-life (assuming quasi first order reactions).

Section 5 presents the results of a sensitivity analysis of ALWAS. Sensitivity to such input parameters as the deposition velocity, settling velocity, scavenging ratio, and degradation half-life has been investigated for a plausible hypothetical scenario of benzo(a)pyrene deposition to a small urban watershed. The results of the sensitivity analysis are interpreted in light of the uncertainty inherent in estimated deposition parameters to conclude that, for some circumstances, the use of estimated, rather than measured, values of the deposition parameters will be adequate to assess effects of airborne contaminants on surface water quality.

1.4 RECOMMENDATIONS

Based on the experience gained in developing ALWAS, it is apparent that there is still much developmental work to do in the field of multi-media modeling. Our comments here will be restricted to ways that developmental work could build upon the ALWAS approach. ALWAS could be made more useful to environmental planners, analysts, and regulators in two directions, expansion to include media or processes not currently considered, and improvement in the description of processes currently simulated. Recommendations are listed in rough order of importance.

1. ALWAS should be extended to simulate groundwater effects more directly. The present version can simulate the surface water quality effects of contaminated groundwater recharge to surface water bodies, but the user must provide independently estimated information regarding groundwater recharge and quality.

2. EXAMS should be extended to simulate the chemical fate of heavy metals. All other components of ALWAS are applicable to heavy metals, but the lack of metals algorithms in EXAMS limits the usefulness of ALWAS for metallic pollutants. Alternatively, DiDOT and NPSDEP could be linked to a model(s) describing the fate of metals in surface water.

3. ALWAS should be improved to account for volatilization from soils, and correspondingly to simulate the atmospheric transport and subsequent redeposition of pollutants volatized from land or water surfaces. EXAMS estimates the rate of volatilization from water surfaces, however, NPSDEP does not contain an analogous description of volatilization from the land surface. Analysis of multi-media impacts of volatile pesticide use, for example, would be hindered by the failure of NPSDEP to describe volatilization. Thus, the first two recommended developments designed to account for the effects of volatilization, would be to extend NPSDEP to estimate volatilization rates from soils. However, as ALWAS is presently implemented, material which volatilizes simply disappears - ALWAS does not track volatilized constituents through the air to subsequent redeposition. Thus, the second recommendation is to modify ALWAS to allow volatilized pollutants to be traced in the air compartment. This problem would be a complex one, because it imposes feedbacks between DiDOT, NPSDEP and EXAMS which would make the current program flow of DIDOT ⟶ NPSDEP ⟶ EXAMS inadequate. It is apparent that the existing flow is only appropriate for substances emitted to air which subsequently are transferred to surface waters.

4. ALWAS requires as input several parameters which control the deposition phenomena, including the settling velocity, deposition velocity, and scavenging ratio. These parameter values have not been determined for many toxic pollutants, so methods for estimating them are given in Section 4.4. The methods presented in that section lend themselves to computerization. It is recommended that software be developed which would allow the user to input only more fundamental, readily available physical chemical properties such as molecular weight, vapor pressure and Henry's law constant rather than the deposition parameters. However, this should remain a user option, since observed values of the deposition parameters would be more reliable than estimates provided by the methods of Section 4.4.

5. Some of the limitations of the existing ALWAS model could be alleviated by substituting a modification of the Agricultural Runoff Model (ARM) for NPSDEP. Substitution of ARM would improve the capability of ALWAS for

simulation of soluble contaminants and the effects of agricultural activities. The resultant improvements would be most effective for the simulation of pesticides. Pesticide simulation would also be facilitated by the implementation of recommendation 3. Substitution of a modified ARM for NPSDEP would not be a major software development effort, because of the very similar structure of the two models.

6. The deposition algorithms used for area sources (including city sources) are not as accurate as those used for point sources. However, area sources may be the greatest contributors to pollutant deposition rates. Exact analytic solutions for area source deposition, analogous to the Ermak solution for point sources, have not appeared in the literature. This fact explains but does not justify the imbalance in accuracy for point vs. area sources. Applied research should be conducted which would identify and quantify the limitations of available algorithms, including the ALWAS approach, and develop better formulations if the existing methods are found to be inadequate.

7. The method by which ALWAS simulates intermittent sources, such as spills or pestidice applications, could be improved to allow for more automatic sequencing of intermittent sources, and to account for longitudinal dispersion in order to describe pollutant "clouds".

8. DiDOT accounts for background air quality, attributable to non-inventoried sources, by use of a single, constant background concentration. This capability should be extended to allow for a matrix of background air quality concentrations which would be dependent on meteorological variables, particularly wind direction.

9. It may be desirable to incorporate land use dependent deposition rates in the transfer of deposition information from DiDOT to NPSDEP. Such a capability was planned for ALWAS, but eventually deleted to improve efficiency in data handling.

10. The NPSDEP/EXAMS link is not realistic in the present version of ALWAS in that the upstream inflows to the EXAMS water body are constant rather than in phase with the NPS flows as would be expected. Capability to process variable upstream inflows had been planned for ALWAS, but was deleted to improve efficiency in data handling.

2.0 OVERVIEW OF AIR LAND WATER ANALYSIS SYSTEM (ALWAS)

2.1 PURPOSE

Statutory requirements for assessment of risks to health and environment have stimulated the need to estimate the movement of chemicals from sources to receptors. Since this involves more than one environmental medium in many cases intermedia transfer rates are critical parameters in gaining some understanding, and control, over the ultimate distribution and fate of pollutants. The determination of the environmental effects of some new chemicals, which would be released into the atmosphere, may critically depend on the water quality effects rather than effects on air quality, because of a greater affinity for water rather than air.

The purpose of ALWAS is to assist in the assessment of water quality problems associated with toxic chemicals released to the atmosphere. The combined effect on surface water quality of toxic releases to several media; including air, non-point source, direct discharge, and groundwater may also be evaluated. Analysis of groundwater impacts is limited since the user must know the rate and quality of groundwater recharge to the surface water body. The model will be useful in determining the most effective strategies for controlling water quality effects of toxic chemicals regardless of the ultimate source, and will be useful in evaluating the potential impacts of new chemicals entering commerce.

2.2 SCOPE AND LIMITATIONS

ALWAS attempts to describe the environmental system depicted in Figure 2-1. The basic flow of the model is from air to land surface to surface water body, with the direct transfer from air to surface water body also accounted for. Atmospheric point sources (smokestacks) and area sources (arising from transportation, land use, residential heating, etc.) can be simulated. Wet deposition, caused by the scavenging of airborne contaminants by precipitation, is simulated as well as dry deposition resulting from such processes as gravitational settling, impaction, or dissolution of gases. Gaseous and particulate contaminants are treated uniformly by the model, consistent with recent findings that many toxic organic contaminants which exist as gases at atmospheric pressure and temperature nonetheless will behave as particulates with respect to deposition processes, since the fraction of the airborne mass which is adsorbed to ambient aerosols is primarily responsible for atmospheric deposition (Eisenreich, et al., 1981; Tucker & Preston 1981). Distinction between contaminants which are primarily particulate, as opposed to those which exist primarily as a vapor, is specified through the gravitational settling velocity, which would be zero for vapors. The dry deposition velocity and the scavenging ratio apply both to vapors and particles.

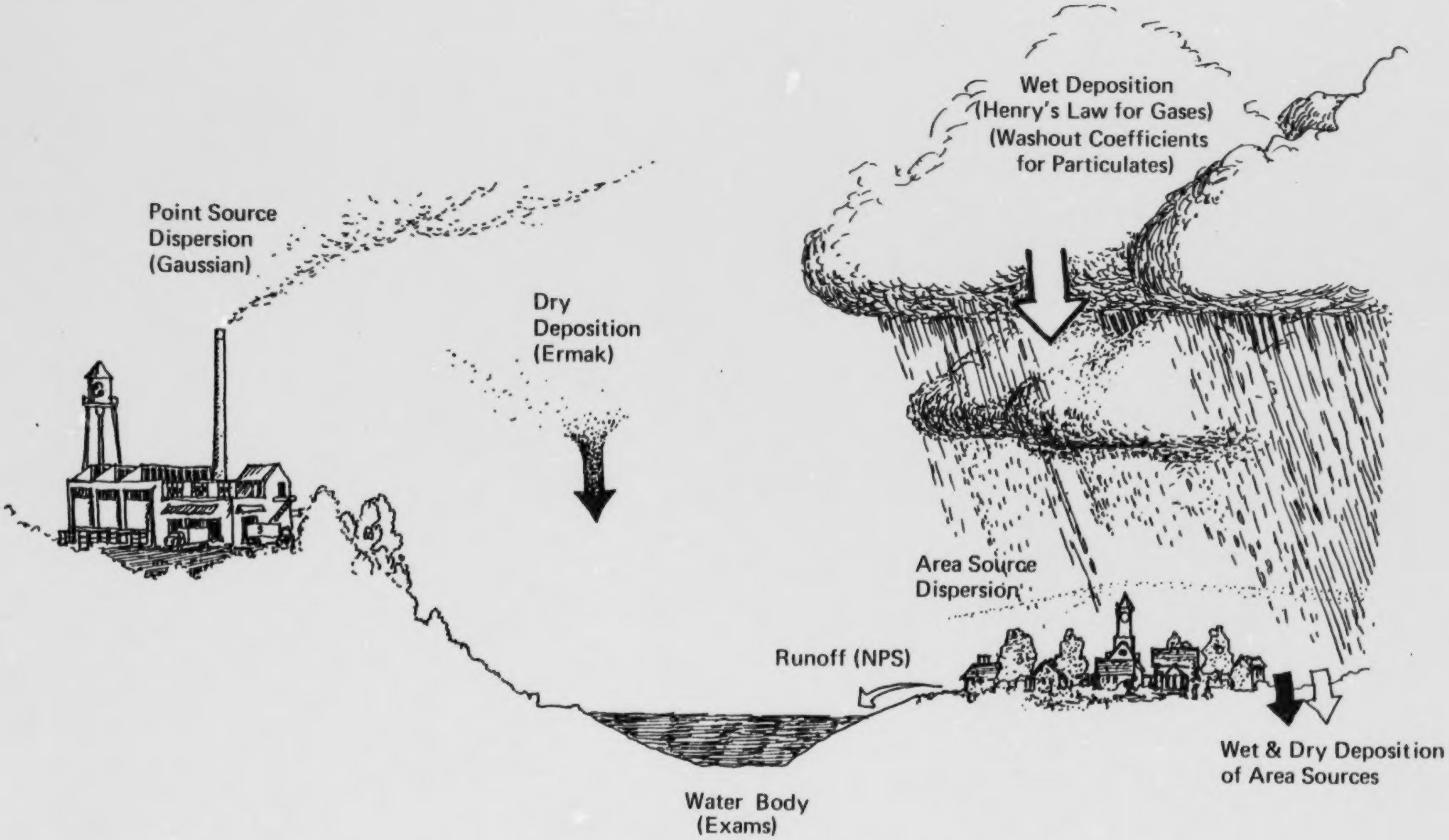

Wet Deposition
(Henry's Law for Gases)
(Washout Coefficients
for Particulates)
Point Source
Dispersion
(Gaussian)
Dry
Deposition
(Ermak)
Area Source
Dispersion
Runoff (NPS)
Wet & Dry Deposition
of Area Sources
Water Body
(Exams)

FIG 2 1

FIGURE 2-1
ENVIRONMENTAL SYSTEM SIMULATED BY ALWAS

The model is designed to be applicable at a wide range of spatial scales ranging from that of a calibrated NPS watershed (limited to about 5 km^2 or (Donigian and Crawford 1976) up to a major river basins covering $10^4 km^2$. Scaling up to $10^4 km^2$ presents technical problems in the atmospheric dispersion model (DiDOT). DiDOT's applicability to large spatial scales is limited because one of its assumptions is questionable at downwind distances approaching 100 km. The assumption is that the wind is uniform, allowing for no meandering within the application area. Even under ideal terrain and meteorological conditions, the assumption of uniform winds may only be valid within about 20 km of the measurement point. If the application area consists of mountainous terrain or a river valley, the scale over which the uniform wind assumption applies may be reduced even further. However, the alternatives to the uniform wind assumption -- multiple station wind data or solution of the Navier-Stokes equations -- are impractical for routine environmental analyses. Both approaches significantly increase computing expenses and would require additional field data not generally available.

ALWAS is a time-dependent model whose fundamental time step is one hour. DiDOT and NPSDEP respond at that time scale while EXAMS interfaces with those two models after temporal averaging over a user-determined number of days. The time dependence of DiDOT is different mathematically from that exhibited by NPSDEP and EXAMS. DiDOT uses a standard air dispersion approach which assumes that steady state conditions exist for each hourly time step. This is possible because the atmosphere responds quickly and the contaminants are resident in the air domain for a short period of time. The approximation is inaccurate under low wind speeds of variable direction, and at very large distances from the source. The approximation is accurate only if pollutants are emitted to the model's air domain and transported through it within the hour, and do not return because of a reversal in wind direction. Theoretically, this limits the applicability in large (10^3-10^4 km^2) watersheds, but the limitation is a practicle constraint only rarely, under the following conditions: a receptor of concern far from a major source is affected adversely by it; this is uncommon since concentrations and deposition rates will be reduced during the long (30-100 km) transport.

On the other hand, both NPSDEP and EXAMS solve differential equations which account for changes in the reservoir of contaminant in their domain from one time step to the next such that current conditions depend on the past history of loading, transport, and transformation. NPSDEP follows a daily time step on days without rain and an hourly time step on rainy days. EXAMS response times are expected to vary widely, depending primarily on the size and flushing time, and the user may specify an appropriate averaging time at integral multiples of one day for loadings provided by DiDOT and NPSDEP. The appropriate averaging time would be less than or approximately equal to the contaminant half life in the aquatic ecosystem as determined by an independent run of the steady state version of EXAMS.

ALWAS is principally designed for organic pollutants which tend to adsorb to particulate matter and are relatively persistent on the land surface. However, ALWAS, or specific combinations of its component sub-models may be used to assess multi-media effects of a broader class of pollutants including heavy metals or relatively hydrophilic organics. For example, NPS runoff and subsurface water quality algorithms do not simulate soluble constituents. However, both DiDOT and

EXAMS account for both soluble and adsorbed contaminants so that these two submodels can be used to determine the effects of direct deposition of hydrophilic organics in water bodies. This feature is useful in lakes whose surface occupies a large fraction of their watershed -- Lake Superior, for example.

EXAMS is not presently designed to simulate the chemistry of metal in aquatic ecosystems. However, air dispersion and deposition of metals is expected to be faithfully simulated by DiDOT, and NPSDEP is suitable for describing land surface processes for relatively insoluble metals. Then DiDOT and NPSDEP could be used to study metals loadings to surface water bodies (from air deposition and runoff) and the relative contribution of various source types to the overall loading, but could not be used to describe surface water quality effects.

Other limitations of ALWAS are principally related to the memory and operating system limitations of the PDP 11/70 computer system for which the model has been designed, and the costs associated with handling the intermediate outputs of DiDOT which must be transferred to NPSDEP and EXAMS. These affect the spatial resolution of deposition rates and area source emissions and the total number of point sources which may be simulated. These limitations may be overcome very simply within the inherent structure of the model by increasing the capacity of various storage arrays if the model were to be implemented on a larger computer or one with virtual addressing capability. However, the cost of storing and handling the larger arrays may limit such a scale-up for many applications, although it has not been the subject of detailed study. The resolution limitations of the software, as it currently exists, may not be a serious drawback for analysis of many deposition-related water quality problems,

An inherent limitation of ALWAS is its inability to appropriately account for volatilization of pollutants from the land or water back to the air. Although EXAMS quantifies the rate of volatilization from the water surface, this contaminant is not fed back to DiDOT as a pollutant air emission; it simply disappears from the model. On the other hand, NPSDEP does not account for volatilization of pollutants from the land surface. The reasons for not accounting for volatilization are severalfold, but principally relate to the technical problems of incorporating feedback into an already complex software structure, particularly in light of the fact that this is a "first-generation" multi-media model. The first-order effect on water quality of deposition of airborne contaminants was of primary interest in this development effort, rather than the second-order effect of volatilization followed by partial redeposition. The limitation suggests that the model be used principally for chemicals having relatively low vapor pressures, although we have not identified a precise range of vapor pressures over which the model is appropriate.

ALWAS may be used to study the environmental impacts of aerial pesticide applications so long as the drift losses, droplet size, effective height of release, and area of application are known. Given this information, ALWAS can be used to evaluate human exposure by inhalation, deposition of pesticide on non-target areas, and the combined impacts from drift and deposition. Use for pesticide evaluation may encounter some problems due to limitations in application of DiDOT algorithms for intermittent air releases, as discussed in 2.3.1. Basically, the restriction is that estimated air concentrations would be valid only near the source for a release of short duration. Consequently, ALWAS may be applied for pesticide application to small study areas (1-10 km^2), but an extension to larger areas would only be possible if large portions of the basin were treated at the same time.

Other, perhaps more serious, drawbacks to the use of ALWAS for pesticides are inherent in NPS which does not account for volatilization and does not simulate the sub-surface movement of soluble contaminants. These drawbacks are discussed further in Section 1.4.

Methods are available for estimating drift losses, droplet size, and effective height of release for pesticidal input to ALWAS. Lawson and Uk (1979), Dumbauld, et al., (1976) Maybank, et al., (1978) and others have performed field studies and developed empirical models for estimating the parameters of droplet release, and these sources may be useful for guidance in developing ALWAS input parameters for application to pesticide spray drift and deposition.

2.3 FEATURES

2.3.1 DiDOT

DiDOT can handle two contaminants in a single run. In order to interface with NPSDEP, one of these contaminants must be total suspended particulates (TSP), while the other one may be any toxic pollutant, either vapor or particulate. Under the recommended mode of operation, the model simulates the behavior of the total airborne contaminant, which may be partially vapor and partially adsorbed to ambient aerosol. DiDOT does not explicitly account for the particle size distribution of particulate contaminant, so the deposition parameters should represent the mass weighted average of the distribution of the parameters. Methods for estimating these parameters are based on the mass median diameter of the particulate contaminant and are presented in Section 4.4.

DiDOT's usage of meteorological and source characteristics data is consistent with standard procedures used in most EPA-supported air quality models. This feature, which makes the model easy to use for those familiar with standard air modeling procedures, is discussed further in Section 2.5. DiDOT is also consistent with standard air modeling procedures in its use of Gaussian dispersion algorithms, plume rise formula presented by Briggs (1973), ASME dispersion parameters, and stability classes based on wind speed and solar radiation. Its method for accounting for limited mixing under an inversion lid, discussed further in Section 4.1 is also consistent with standard approaches.

Where DiDOT diverges from standard air quality models is in its deposition algorithms. The dispersion algorithms utilized in DiDOT represent a synthesis and application of current research and mathematical analysis of the deposition process. In order to describe the dry deposition process from point sources, DiDOT applies the rigorous analytical solution derived by Ermak (1977) which accounts for both gravitational settling and surface depletion. Available models which incorporate deposition phenomena use the source depletion method. The difference between these, and a critique of the source depletion approach has been presented by Horst (1977).

The surface depletion model correctly accounts for the fact that pollutants are deposited at the ground, thus reducing ground-level concentrations relative to the values which would be observed if deposition were not occurring, and leading to a vertical gradient of concentration at the surface. The source depletion model, on the other hand, preserves the Gaussian shape for the vertical concentration profile. It accounts for upwind deposition by adjusting the source emission rate, i.e., subtracting the surface integral of the upwind deposition flux from the emission rate. This approximation adjusts the source strength artificially by adjusting the volume of material in the plume in proportion to the concentration, rather than a surface depletion model in which deposition occurs at the ground. The effect of various deposition approximations on the shape of the vertical concentration profile is shown in Figure 2-2. As is indicated, the assumptions of no deposition or source depletion result in overestimation of ground level concentrations. For the source depletion model, this leads to an overestimate of the deposition rate as well.

Ermak's (1977) surface depletion solution is applicable only for a point source with unlimited vertical mixing (no interaction with inversion lid). The solution is applicable near the source, but once the vertical dispersion of the contaminant plume begins to be limited by an inversion layer, the solution is no longer valid. This limitation also applies to commonly used Gaussian dispersion equations which do not account for deposition. In air dispersion methodology, the common solution to this problem is to define a downwind distance X_L at which the interaction with the inversion lid begins to be significant. At twice this downwind distance $2X_L$, the concentration is assumed to be vertically uniform (Turner, 1970). Between X_L and $2X_L$ the concentration is interpolated logarithmically. This approach has been modified in DiDOT to also account for deposition from X_L to $2X_L$ by applying Ermak's result for total upwind deposition at $2X_L$, which is then used in the standard source depletion method to estimate the concentration at $2X_L$. Beyond $2X_L$ the source depletion method is used because the details of profile distribution become unimportant.The source depletion approach are thus introduced to the model beyond $2X_L$. Under typical meteorological conditions $2X_L$ will rarely be less than 10 km and may be much more than 200 km. Under extremely rare, and transitory, atmospheric conditions, $2X_L$ may be as little as 2 km. Thus, for most typical meteorological conditions, the inherent inaccuracy of the source depletion model will affect only a small fraction of DiDOT's domain, at large downwind distances where ground level concentrations and deposition rates are relatively small.

Wet deposition processes are simulated via the scavenging ratio approach recommended by Gatz (1976). The basic assumption of the scavenging ratio approach is that the rainfall concentration is proportional to the ground level air concentration. Theoretically, several precipitation scavenging processes should not follow this pattern. These include "in-cloud" scavenging orrainout, and the irreversible scavenging of an elevated plume. Nonetheless, for many contaminants, under diverse meteorological conditions, the assumption proves relatively accurate (Gatz, 1978). The scavenging ratio for toxic organic pollutants can be estimated to within an order of magnitude, following procedures presented in Section 5.4 (Tucker, Lyman & Preston, 1982). During rainfall, DiDOT does not account for upwind wet deposition in determining the ground level air concentration.

Although DiDOT is primarily designed to simulate dispersion from continuous sources, it can be adapted to simulation of an intermittent source such as a chemical spill, or a pesticide application. This is accomplished by performing multiple runs with a selected subset of meteorological data. The DiDOT post-processor, which manipulates DiDOT output in order to create NPSDEP and EXAMS input data sets, is designed to merge such multiple runs -- which would most likely consist of one or more continuous sources in one run, and one or more intermittent sources in an auxiliary run -- to determine total deposition rate inputs for NPSDEP and EXAMS. The air dispersion algorithms of DIDOT are technically valid for intermittent sources only if the downwind source-to-receptor distance is less than the product of the source emission duration and the wind speed. Thus, under calm conditions (U = 2m/sec), an intermittent source of three-hour duration would be accurately simulated by DiDOT to a range of 22 km. Farther downwind, the effect of source intermittency must be accounted for by specific consideration of longitudinal dispersion, which may be safely neglected for continuous sources.

2.3.2 NPSDEP

NPSDEP is the result of relatively minor modifications to NPS. The features of NPS are described in detail by Donigian and Crawford (1978), and are briefly summarized below.

NPS is a continuous simulation model that represents the generation of non-point source pollutants from the land surface. The model simulates surface and subsurface hydrologic processes, snow accumulation and melt, sediment generation, pollutant deposition [note that deposition is referred to as accumulation by Donigian and Crawford (1978)], and pollutant transport for any specified period of meteorological data. NPS inputs include parameters which allow the user to adjust the model to a specific watershed. Most of these parameters are specified by physical (observable) characteristics of the watershed; however, several of the parameters cannot be determined from observations, and the model must be calibrated before application to any watershed. NPS permits simultaneous simulation of five user-specified contaminants. These contaminants are assumed to migrate with sediment entrained in surface runoff. Consequently, subsurface water is assumed to be clean. This constraint limits the validity of the model to pollutants which are strongly associated with sediments including hydrophobic organics and heavy metals. The contaminants are assumed to be conservative on the land surface, which implies that the model is most useful for non-volatile, persistent pollutants. For volatile, degradable contaminants, NPS loadings outputs may be useful as an upper bound.

NPS allows monthly variations in land cover, pollutant deposition (accumulation) and pollutant removal (e.g., by street cleaning).

The only significant modifications to NPS which are incorporated in NPSDEP relate to the deposition processes simulated by DiDOT. Where NPS allowed monthly variations in deposition rates, NPSDEP accepts daily variations. NPS permitted land-use-dependent ratios between contaminant and sediment (potency factors) which were assumed to be the same for eroded sediment and deposited sediment, but NPSDEP allows two separate potency factors per land use -- one for eroded sediment from the land surface and one for the deposited sediment from DiDOT.

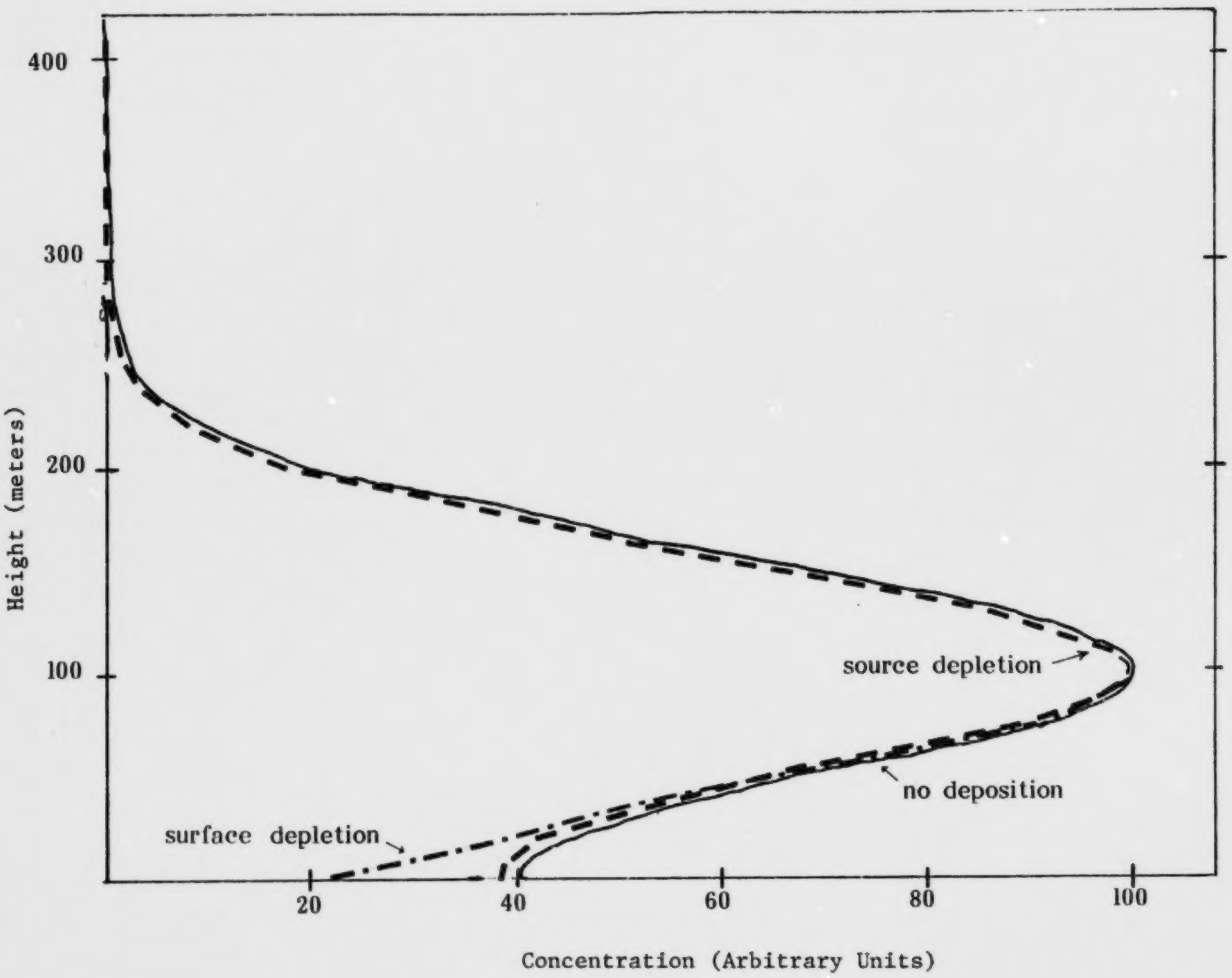


FIGURE 2-2

COMPARISON OF VERTICAL CONCENTRATION PROFILES
IN ALTERNATIVE MODELS

Monthly variations in the potency factors are permitted. Thus, NPSDEP can simulate the combined impacts of deposited contaminant and some independent source of non-point pollution as shown in Figure 2-3.

NPSDEP also incorporates a simple method of accounting for contaminanted precipitation in contrast with NPS which assumes the precipitation is pure water. In NPSDEP, contaminant associated with precipitation is assumed to pass, unmodified, through the watershed land surface to the stream.

2.3.3 EXAMS

The dynamic version of EXAMS-AETOX1, used by ALWAS, is modified from the version currently supported by EPA's Athens Environmental Research Laboratory. EXAMS describes the transport and chemical transformation of organic chemicals in surface water bodies. It is a multiple compartment, or cell, model, where the cells are linked by advection and turbulent exchange. Cells may contain mostly water (littoral, epilimnion, hypolimnion, etc.) or mostly sediment (benthic), with different input requirements. However, each cell consists of water, sediment, and biomass in variable proportions.

Chemical processes are simulated on the basis of either equilibrium partitioning or first order kinetic processes. Equilibrium partitioning among a (5 x 3) matrix of possible forms is established for up to five (5) ionic forms (valences of -2, -1, neutral, +1, +2) and between dissolved, sediment adsorbed, and biosorbed phases. Processes simulated by first order kinetics include hydrolysis, oxidation, volatilization, microbial transformation, and photolysis.

The dynamic version of EXAMS-AETOX1 used by ALWAS determines the dynamic response to time-varying loading rates. This modification is discussed further in Section 4.2.3.

2.4 INPUT/OUTPUT

2.4.1 INPUTS

ALWAS requires inputs which may be separated into six basic categories:

- meteorological data,
- pollutant source data,
- pollutant fate properties,
- calibration data for NPSDEP,
- characteristics of the physical environment (primarily for NPSDEP and EXAMS), and
- interfacing data.

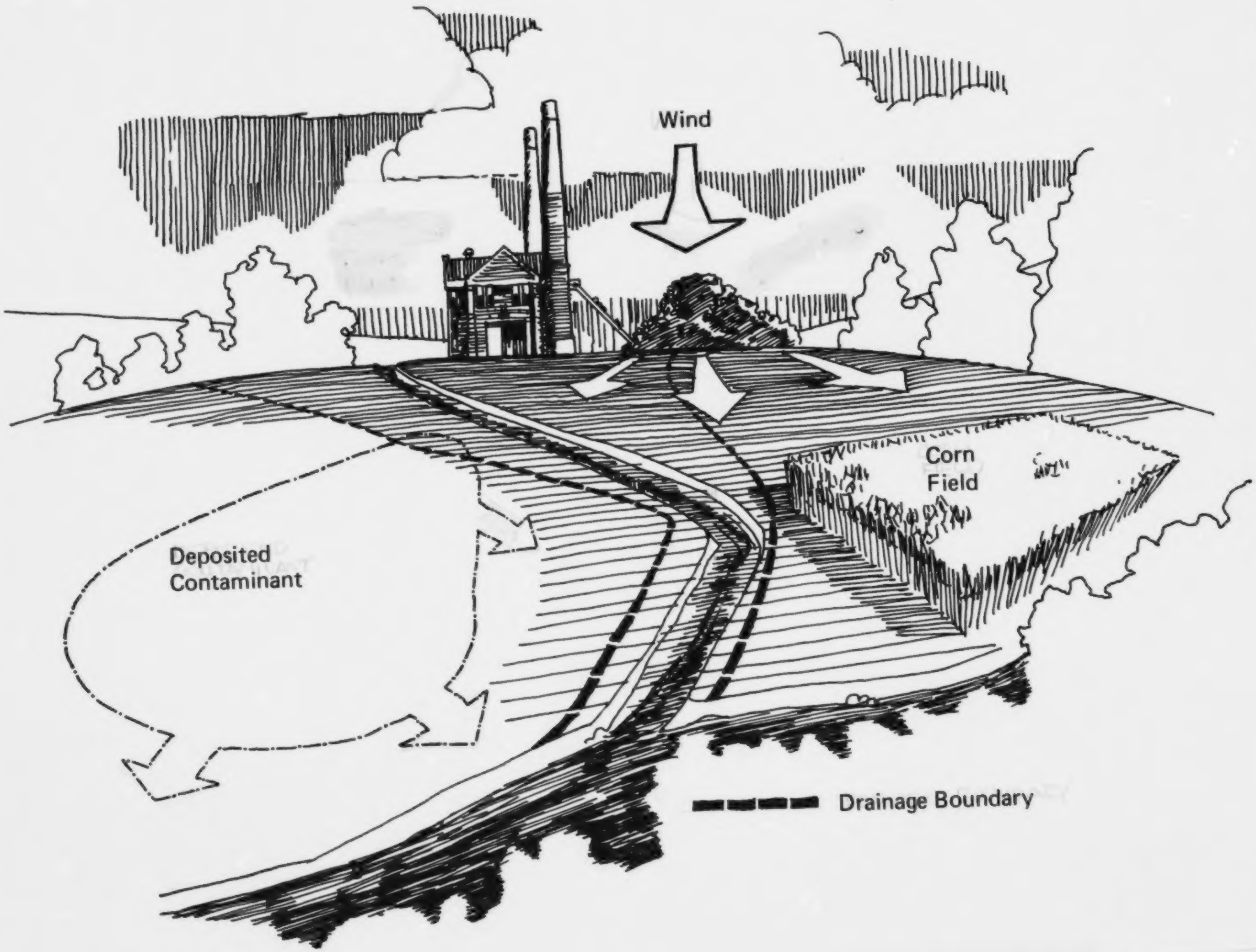

Wind
Corn
Field
Deposited
Contaminant
Drainage Boundary

FIGURE 2-3. NPSDEP APPLICATION IN A WATERSHED

Meteorological data: each sub-model of ALWAS requires meteorological data of different kinds. Both DiDOT and NPSDEP require hourly meteorological data in formats compatible with standard reporting formats of the National Weather Service.

Pollutant source data: DiDOT and EXAMS require data characterizing sources of the pollutant to the environment DiDOT treats three kinds of pollutant sources: point sources, traditional area sources, and distant city-sized area sources. Traditional area sources represent air emissions from transportation, small institutional and commercial sources, residential heating or solvent use, land use activities -- in short, all small sources which are too numerous or otherwise inconvenient to inventory individually. Distant cities are treated by a virtual point source approach. The data required to characterize sources in DiDOT are typical of those required by other air quality models, and are available in standard state-maintained emissions inventories.

Pollutant fate properties: chemical specific inputs are required primarily by DiDOT and EXAMS. DiDOT uses the settling velocity, deposition velocity, scavenging ratio, and atmospheric degradation half-life. These, in turn, may be estimated using procedures presented in Section 4.4, from knowledge of particle size, Henry's law constant, vapor pressure, and molecular weight, or other fundamental properties. EXAMS requires similar inputs, solubility in water, octanol/water partition coefficient, rates of hydrolysis, photolysis, and oxidation.

Calibration data for NPSDEP: NPSDEP requires calibration data -- time series of water flow and quality at the discharge point of an upland watershed. The availability of such data is critical to the application of ALWAS. This topic is discussed further by Donigian and Crawford (1976).

Characteristics of the physical environment: NPSDEP requires as input various physical features of the watershed -- area, length of overland flow path, slopes, elevation, land use characteristics, portions of watershed which are pervious and impervious, soil characteristics, etc.

EXAMS relies extensively on a user-specified description of the water body, since it does not solve equations which describe its hydraulics. If multiple cells are specified, the user must quantify the advective and turbulent exchanges between cells. This demands an in-depth knowledge of flow patterns in the water body, which may be achieved via field studies and/or independent numerical modeling using models which solve the equations of motion. In lieu of such information, the user may rely on the descriptions of typical aquatic ecosystems contained as a default input data file within the EXAMS data base.

Interfacing data: within this category are the geometrical relationships between the DiDOT receptor locations (points at which deposition rates are calculated), NPSDEP catchments, and EXAMS cells. Each of the models has unique ways of describing specific locations within its domain, and these are interrelated in a straightforward fashion.

As should be clear from the foregoing discussion, the data requirements are extensive, leading to significant costs in data gathering, formatting, and handling prior to implementation of DiDOT. However, most of these inputs are routinely available from environmental data banks, including those maintained by EPA, the National Climatic Center, and the U. S. Geological Survey.

2.4.2 OUTPUTS

The fundamental ALWAS outputs are the surface water quality outputs presented by EXAMS and described in detail by Burns, et al., (1980). These include the total contaminant concentrations, as a function of time, in each EXAMS cell, and the partitioning of that total contaminant among various ionic, dissolved, sediment adsorbed, and biosorbed forms. DiDOT also presents summary air quality outputs including long-term average concentrations over the watershed and the extreme hourly and daily average concentration. NPSDEP provides flow, sediment transport, and toxicant loading outputs at monthly and annual intervals, as well as short term output for major storms. NPSDEP output is described by Donigian and Crawford (1976).

3.0 TECHNICAL BASIS OF ALWAS

3.1 MODEL FOR AIR DISPERSION AND DEPOSITION OF TOXICS (DiDOT)

Calculation methods were developed to estimate surface level concentrations in air and air-to-surface transfer of toxic substances. Inputs include meteorology data, source emissions inventories, and pertinent properties of the toxic substance. Outputs are air concentrations varying with time and location, and flux rates from the atmosphere to the soil or water.

Source inventories generally do not include distributed volatilization from water and soil, but rather the inputs due to anthropogenic sources. If emissions from natural media are not negligible and are susceptible to independent estimation, they can be incorporated in the same formalism as the anthropogenic sources. This is to avoid the need to run air, runoff, and aquatic models simultaneously. Coupling within a watershed defined by the aquatic and runoff models will be from air-to-surface, but will not occur to a significant extent in the reverse direction because the residence time of air over the watershed is very short compared with that associated with the movement of water or soil within the watershed. Thus, failure to effect a feedback link, wherein volatilization affects air concentrations which then affect deposition rate, will not seriously affect the water quality results, but may limit the accuracy of predicted air concentrations.

In general, a goal for the air model formulation is to maintain consistency with the two models in level of approximation, in the classes of chemicals that can be described, and in resolution - both in time and space. The Guassian dispersion formulations and first-order chemical degradation are appropriate assumptions for the air model under these criteria. (EXAMS has first-order fate processes, but has a minimum of spatial differentiation of conditions.) Organic chemicals are the primary emphasis because inorganic chemistry is not treated either in the runoff or aquatic models. The model will also be suitable for contaminants which are inert in the atmosphere. This is not very restrictive because 114 of the 129 priority pollutants are organic, and all of the action list chemicals from the Interagency Testing Committee are organic. Various forms of the continuous source Gaussian models will be chosen on the basis of source geometry, receptor geometry, and source/receptor distance. Transfer processes of fallout, dry deposition, and washout are included. In the following, we will first describe a standard point source algorithm and subsequently the modifications to that algorithm which are required to describe deposition phenomena.

3.1.1 POINT SOURCE MODEL WITHOUT DEPOSITION

This formulation is similar to that of the single source (CRSTER) model in that it is a steady-state Gaussian plume. It accounts for a first order atmospheric degradation by the use of an exponential decay factor. the surface concentration downwind from the source is given by Sutton (1953):

$$\chi = \frac{Q}{2\pi\sigma_y\sigma_z u}\exp\left[-\frac{1}{2}\left(y/\sigma_y\right)^2-\frac{1}{2}\left(H/\sigma_z\right)^2\right]\exp\left(-t/\tau\right) \tag{3-1}$$

where χ = surface concentration. (gm^{-3})
Q = source emission rate. (gs^{-1})
σ_y = crosswind dispersion parameter. (m)
σ_z = vertical dispersion parameter (m)
u = wind speed (ms^{-1})
y = crosswind distance from plume centerline (m)
H = effective plume height (m)
τ = characteristic chemical time (s)

The dispersion parameters are calculated from the formulas (ASME, 1968):

$$\sigma_y = ax^b \; ; \; \sigma_z = cx^d \tag{3-2}$$

where the coefficients, a, b, c, and d are specified in Table 3-1.

3.1.2 TREATMENT OF EFFECTIVE STACK HEIGHT

In order to calculate the effective plume height H the following formula is used:

$$H = h_s + h_p - C \quad \text{if} \quad H > \tfrac{1}{2}(h_s + h_p) \tag{3-3}$$

TABLE 3-1

ASME COEFFICIENTS AND EXPONENTS USED TO CALCULATE HORIZONTAL AND VERTICAL PLUME DISPERSION PARAMETERS

Stability Class	Values			
	a	b	c	d
1. Very Unstable	0.40	0.91	0.40	0.91
2. Unstable	0.36	0.86	0.33	0.86
3. Neutral	0.32	0.78	0.22	0.78
4. Stable	0.31	0.71	0.06	0.71
5. Inversion-Very Unstable Below	0.40	0.91	0.40	0.91
6. Inversion-Unstable Below	0.36	0.86	0.33	0.86
7. Inversion-Neutral Below	0.32	0.78	0.22	0.78

Source: ASME (1968)

19

where C = low exit velocity correction for downwash
h_p = plume rise above the release point
h_s = stack height

If the above inequality is not satisfied, the value of H is set at $(h_s + h_p)$; that is

$$H = \tfrac{1}{2}(h_s + h_p) \text{ if } H \leq \tfrac{1}{2}(h_s + h_p). \tag{3-4}$$

The downwash correction is employed when the vertical stack exit velocity is less than 1.5 times the horizontal windspeed. It is given by: (Gifford, 1978)

$$\begin{aligned} C &= 3(1.5 - W_o/u)D \quad \text{for } W_o < 1.5u \\ C &= 0 \quad \text{for } W_o \geq 1.5u \end{aligned} \tag{3-5}$$

where C = downwash correction
D = inside stack diameter
u = windspeed
W_o = vertical exit velocity of plume

The plume rise H_p is estimated by the method of Briggs (1970):

$$\begin{aligned} h_p &= \frac{1.6 F^{1/3} x^{2/3}}{u} \quad \text{for } x < 3.5x^* \\ h_p &= \frac{1.6 F^{1/3} (3.5x^*)^{2/3}}{u} \quad \text{for } x \geq 3.5x^* \end{aligned} \tag{3-6}$$

In unstable and neutral conditions where

F is the buoyancy flux in ($m^4\,s^{-3}$)

$$F = \left[g W_o D^2 (T_e - T_a)/4 T_e \right] \tag{3-7}$$

χ^* is the downwind distance at which atmospheric turbulence dominates plume mixing (m)

$$\begin{aligned} \chi^* &= 14F^{5/8} \text{ if } F \le 55 m^4 s^{-3} \\ &\quad 34F^{2/5} \text{ if } F > 55 m^4 s^{-3} \end{aligned} \tag{3-8}$$

3.5χ^* is the downwind distance at which the plume height stabilizes (m)

g = Gravitational acceleration

T_e = Exit temperature of effluent - (°K)

T_a = Ambient temperature (°K)

For stable conditions:

$$h_p = \frac{1.6F^{1/3}\chi^{2/3}}{u} \text{ for } \chi < 2.4us^{1/2} \tag{3-9}$$

$$h_p = 2.9\left(\frac{F}{uS}\right)^{1/3} \text{ for } \chi \ge 2.4us^{1/2}$$

Where S is the atmospheric stability parameter defined by:

$$S = \frac{d\theta}{dz} \quad \frac{g}{T_a} \tag{3-10}$$

where

θ = Potential temperature*

z = Height

For plume rise in a calm atmosphere, i.e., $u < 3$ mph the following equation is used:

$$h_p = 5F^{1/4}S^{-3/8} \tag{3-11}$$

---

*The temperature which a parcel of air would have if it were expanded or compressed adiabatically from its existing pressure and temperature to 1000 millibars.

These parameters are appropriate for hourly concentrations because the σ_y and σ_z parameters were determined for this averaging time. Equation (1) is used with stability classes 1 through 4. Class 1 corresponds to Pasquill (1961) categories A and B; class 2 corresponds to C, Class 3 to D; and Class 4 to E and F. The selection of stability class follows the method of Table 3-2.

Classes 5, 6 and 7 are used when the upper extreme of the plume is near the mixing depth or inversion base elevation. It is possible to use a complicated trapping formula for this case, but a simpler approach is to neglect trapping until

$$\sigma_z = 0.47L \text{ (at } X = X_L\text{)}$$

where L is the mixing height and x is the downwind distance. At $X \geq 2X_L$ the approximation (Turner, 1967) that we will use is the uniform vertical mixing has taken place so that Equation (1) is replaced with:

$$\chi = \frac{Q}{\sqrt{2\pi}\ \sigma_y L u} \exp\left[-\frac{1}{2}\left(y/\sigma_y\right)^2\right] \exp\left(-t/\tau\right) \tag{3-12}$$

For $X_L < X_L < 2X_L$ a linear interpolation of log χ with log x is made between $\chi(X_L)$ determined from Equation (1) and $\chi(2X_L)$ determined from Equation (3-12).

Continuing downwind the solution for horizontal plume spread should follow the long range (100-1000 km) correlations presented by Bigg, et al (1978). Using measured data reported in six different literature sources, they fitted a linear regression formula with an exponent of 0.88 and found a sample correlation coefficient of 0.91. This exponent is approximated well by the range of b-values for stability classes 5, 6, and 7 in Table 3-1 considering the wide (plus or minus a factor of 3) scatter band in the data used by Bigg, et al (1978). Thus, we will use the ASME parameters for the long-range transport cases as well as for the near-field.

3.1.3 POINT SOURCE MODEL INCORPORATING DRY DEPOSITION PROCESSES

When it's not raining, the intermedia processes to be treated for air/surface transfer are settling and dry deposition. The first is the result of the gravitational settling of particulate matter, the second arises from absorption by the surface.

Fallout and dry deposition have been treated in an exact solution to the species budget equation by Ermak (1977). The equations describing contaminant dispersion and deposition from a point source are:

$$u \frac{\partial \chi}{\partial x} = K_y \frac{\partial^2 \chi}{\partial y^2} + K_z \frac{\partial^2 \chi}{\partial z^2} + W_s \frac{\partial \chi}{\partial z} \qquad (3\text{-}13)$$

where

K_y & K_z are turbulent diffusivities in the transverse and vertical directions

W_s is the gravitational settling velocity

Boundary conditions are:

$$\chi(0,y,z) = \frac{Q}{u} \delta(y)\, \delta(z-h)$$ point source (3-14)

where the Dirac delta functions are defined

$$\delta(y) = 0 \,, \quad y \neq 0 \quad ; \qquad \int_{-\varepsilon}^{\varepsilon} \delta(y)\, dy = 1$$

$$\chi(x, \pm\infty, z) = 0$$
$$\chi(x, y, \infty) = 0$$

no pollution far (3-15)
from source (3-16)

$$\left[K \frac{\partial \chi}{\partial z} + W_s \chi \right]_{z=0} = W_d \chi \Big|_{z=0}$$

deposition rate (3-17)
proportional to
ground level
concentration

and defining the dispersion coefficient, σ :

$$\sigma^2(x) = \frac{2}{u} \int_0^x K(x')\, dx' \qquad (3\text{-}18)$$

TABLE 3-2

SELECTION OF STABILITY CATEGORIES

Surface Wind Speed (at 10m) m sec-1	Day Incoming Solar Radiation Strong	Moderate	Slight	Night Thinly Overcast or 4/8 Low Cloud	<3/8 Cloud
<2	A	A-B	B		
2-3	A-B	B	C	E	F
3-5	B	B-C	C	D	E
5-6	C	C-D	D	D	D
>6	C	D	D	D	D

Source: Turner (1967)

The solution to this system of equations is:

$$\chi(x,y,0) = \frac{Q e^{-t/\tau}}{2\pi \sigma_y \sigma_z u} \exp\left[-\tfrac{1}{2}\left(\tfrac{y}{\sigma_y}\right)^2\right] \tag{3-19}$$

$$\exp\left[-\frac{w_s H}{2K} - \frac{w_s^2 \sigma_z^2}{8K^2}\right]$$

$$\left\{2 \exp\left[-\frac{1}{2}\left(\frac{H}{\sigma_z}\right)^2\right]\right.$$

$$\left. - \sqrt{2\pi}\, \frac{w_1 \sigma_z}{K} \exp\left[\frac{w_1 H}{K} + \frac{w_1^2 \sigma_z^2}{2K^2}\right] \operatorname{erfc}\left(\frac{w_1 \sigma_z}{\sqrt{2}\, K} + \frac{H}{\sqrt{2}\, \sigma_z}\right)\right\}$$

where K = vertical diffusivity at large distances and $w_1 = w_d - w_s/2$. The average vertical diffusivity at large distances can be estimated from:

$$K \sim \sigma_z^2/2t \simeq \frac{\sigma_z^2 u}{2x} \tag{3-20}$$

for the purposes of substitution in Equation (3-19).

Note that the Ermak solution may be interpreted as a modification of the Gaussian expression of Equation (3-1). The first and third exponential terms are found in Equation (3-1). The second exponential term quantifies the gravitational settling of particulates from their effective release height, H, under the influence of vertical diffusion during their travel. The last exponential term describes the effect of pollutant deposition on the ground level concentration, while the erfc (complimentary error function) term represents the effect of upwind deposition. The erfc term appropriately accounts for the fact that pollutants deposited upwind have been removed from the air at its lower boundary. Each upwind location at which deposition has occurred acts like a point sink, or negative source.

Using the same analysis, we calculate the net deposition rate from the two processes by integrating the vertical flux, J, of the pollutant into the surface:

$$J_z(x,y,0) = -K\frac{\partial \chi}{\partial z} - w_s \chi \qquad (3\text{-}21)$$

from the source point to the downwind distance, X , in the form:

$$N(x) = -\int_0^x d\xi \int_{-\infty}^{\infty} J_z(\xi,\eta,0)\,d\eta \qquad (3\text{-}22)$$

where N is the total amount of pollutant deposited between the source and x per unit time. Carrying out the operations indicated by Equations (3-21) and (3-22) with the concentration expression in Equation (3-19) yields:

$$N(x) = Q e^{-t/\tau}\left[1 - \frac{1}{2}\,\mathrm{erfc}\left[\frac{w_s\sigma_z}{2\sqrt{2}\,K} - \frac{H}{\sqrt{2}\,\sigma_z}\right] + \frac{w_d}{2w_2}\right.$$

$$\times \exp\left(\frac{w_s H}{\sigma_z}\right)\mathrm{erfc}\left[\frac{w_s\sigma_z}{2\sqrt{2}\,K} + \frac{H}{\sqrt{2}\,\sigma_z}\right] \qquad (3\text{-}23)$$

$$\left. - \frac{w_1}{w_2}\exp\left[\frac{w_1 H}{K} + \frac{w_2\sigma_z^2}{2K^2}\right]\mathrm{erfc}\left[\frac{w_1\sigma_z}{\sqrt{2}\,K} + \frac{H}{\sqrt{2}\,\sigma_z}\right]\right]$$

where $w_2 = w_d - w_s$. As indicated for practical cases, (Pasquill, 1974) W_d is somewhat greater than W_s so that Equation (3-23) will not be indeterminate.

The Ermak solution, represented by Equations (3-19) and (3-23), is only valid then there is unlimited vertical mixing. As discussed in 4.1.2, this is practically true when $\sigma_z < 0.47L$. By definition, $\sigma(X_L) = 0.47L$, so that Equations (3-19) and (3-23) are valid for $X < X_L$. X_L may range from 1 km to hundreds of km under extreme meteorological conditions, but is rarely less than 10 km. Beyond X_L the plume begins to interact with the mixing lid to the extent that ground level concentrations and deposition rates would be affected. A standard rule of thumb, used as the basis for approximation in air dispersion

modeling practice, is that beyond $2X_L$ the concentration is vertically uniform, and that assumption is adopted in DiDOT. However, this approach doesn't provide an indication of how much contaminant has been deposited from X_L to $2X_L$. In DiDOT, it is assumed that Eq. (3-23) is valid at $2X_L$.
The settling velocity is obtained from Stokes Law:

$$w_s = \frac{\rho_s g d^2}{18\mu} \tag{3-24}$$

where d = effective particle diameter
g = gravitational acceleration
ρ_s = particle density
μ = dynamic viscosity by air

The deposition velocity, sometimes referred to as exchange rate, relates the air-to-surface flux of a contaminant to its ambient concentration at instrument height (3-10 m).

This quality may be affected by aerodynamic characteristics, surface properties, and the physico-chemical properties of the pollutant. There are numerous pollutants for which physico-chemical properties are not controlling factors. These include fine particulates (smaller than 10 micrograms) and perfectly absorbing gases. For such contaminants the deposition velocity is determined by the rate at which atmospheric turbulence brings the contaminant to the surface: Each particle or molecule once brought to surface stays there.

DiDOT inputs include a nominal deposition velocity and the gravitational settling velocity. By comparing these two parameters to predefined limits, DiDOT determines whethers physico-chemical propeties are significant. If not, it estimates the true deposition velocity as a function of meteorological and surface conditions. The rationale is as follows:

Peterson (1977) summarizes available information regarding the variability of the deposition velocity. He concludes, as does Slinn, et al., (1978), that atmospheric turbulence brings contaminants to the surface at a rate ranging from 0.1 to 10 cm/sec, and averaging about 1 cm/sec over longer time intervals (weeks to years). The gravitational settling velocity is about 10 cm/sec for a 20μm particle and increases as the square of the diameter for larger particles. For particles much larger than 20μm, the settling velocity is much greater than the exchange rate caused by turbulence, so settling predominates, and the deposition velocity equals the settling velocity. DiDOT overrides an input nominal deposition velocity and uses the settling velocity if the settling velocity is greater than 20 cm/sec.

On the other hand, for sparingly soluble or unreactive gases, the deposition velocity may be constrained by the surface. Hill (1971) and Hill & Chamberlain (1974) have concluded that interaction with surface water (even for deposition on vegetation or soil) is the most important interaction for many pollutants.

If a pollutant is insoluble or unreactive in water, then the surface may be at or near saturation with respect to the air concentration. In this case, the contaminant is reflected from the surface, and the net flux is much less than the rate at which molecules are brought to the surface by turbulent air currents: The transfer rate is liquid phase controlled (again assuming that water is analogous to other surfaces).

Such contaminants are characterized by slow water reaction rate constants and large Henry's law constants. For these contaminants the input deposition velocity will be less than 0.1 cm/sec., and DiDOT applies this low rate regardless of meteorological conditions. The basis for estimating very low deposition velocities based on the contaminant's physico-chemical properties is discussed in 4.4.

When the gravitional settling velocity $W_s < 10$ cm/sec., and the nominal deposition velocity $0.1 < W_d < 10$ cm/sec., then the deposition rate, proportional to the actual deposition velocity, is dependent on meteorological and surface conditions and defined as follows:

$$W_d = W_a + W_s \tag{3-25}$$

W_a is the exchange rate associated with turbulent exchange,

$$W_a = \frac{U_*^2}{U} \left(1 + \frac{U_*}{UB} \right)^{-1} \tag{3-26}$$

where U_* is the friction velocity,

U the windspeed, and

B the Stanton number $7 < B^{-1} < 10$

be definition:

$$U_* = \sqrt{\frac{\tau_o}{\rho_a}} = U \sqrt{C_d} \tag{3-27}$$

where τ_o is the surface wind stress,

ρ_a the density of air, and

C_d the aerodynamic drag coefficient

By definition:

$$\frac{kz}{U_*} \frac{\partial U}{\partial z} = \phi_m (z/L) \tag{3-28}$$

where L is the Monin-Obukhov length,
k is the von-Karman constant, and
z is the vertical coordinate, and
$\phi_m(z/L)$ is a self-similar vertical velocity gradient.

Experimental data (Businger, et al., 1971) gives ϕ_m as a polynomial in (z/L).

Thus,

$$U_* = \frac{kU}{\int_{zo}^{zw} \frac{\phi m}{z} \, dz} \tag{3-29}$$

where z_o is the surface roughness length--a required input which incorporates surface features, and
z_w is the instrument height (3-10m)

3.1.4 Area Source Model

This approach depends on the Monin-Obukhov length, L, which is difficult to derive from surface weather observations, but can be related approximately to the stability class (an hourly input derived from surface weather observations).

Typical values of the Stanton number, B, are given in Table 3-3. The surface roughness for some other surface types is indicated in Figure 3-1. The deposition velocity is used for both point and area sources as described below.

For estimating surface concentrations at locations downwind of an extended area source such as an urban complex, we shall use an approximation based on Lucas (1958). This averages the ground level concentration over an area as a function of uniform source strength $\overline{Q}(gm^{-2}\ 5^{-1})$, mixing height (L), wind speed (u) and downwind distance (x). Neglecting lateral diffusion the ground level centerline concentration may be written as:

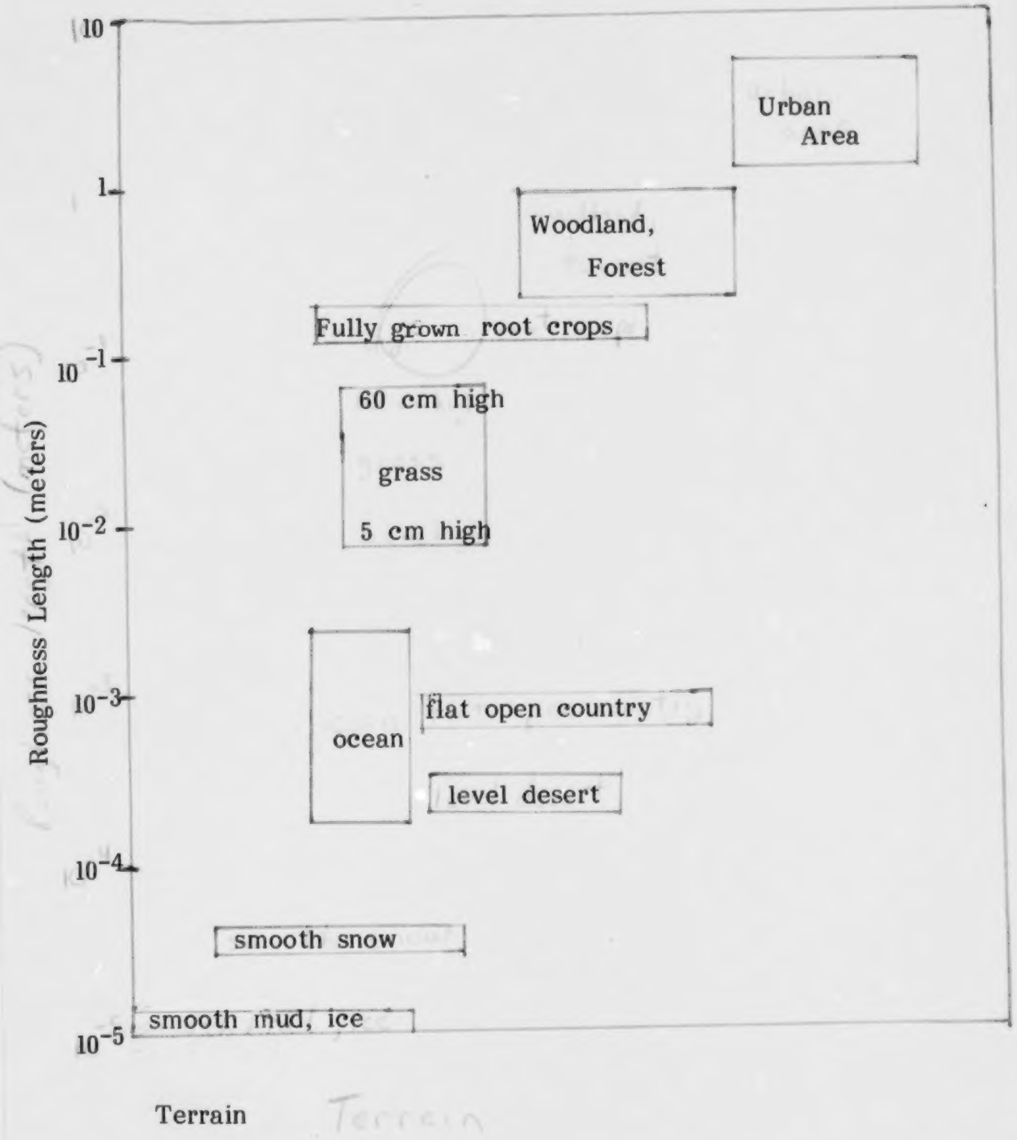
TABLE 3-3

MEAN VALUES AND STANDARD ERRORS (S.E.)
OF THE RECIPROCAL STANTON NUMER. B^{-1}

Surface	Roughness height, Zo. (m)	B^{-1} for u_*(m/s) 0.25	0.5	1.0	S.E. of B^{-1}
Grass (~7cm high)	0.007		7.7		0.4
Short grass	0.002	7.9	8.7	10.3	0.6
Artificial grass	0.01	7.0	8.0	10.1	0.3
Towelling	0.00045		8.0		0.6
Rough glass	0.0002	28.5	30.7	35.3	1.4

Source: Pasquill (1974)

FIGURE 3-1
VARIATION OF ROUGHNESS LENGTH WITH TERRAIN FEATURES



Sources: Seinfeld (1975)
Pasquill (1974)
Slade (1968)

$$\chi(x,0,0) = \int_{0}^{x}\int_{-B}^{B} \frac{2\bar{Q}}{2\pi\sigma_y\sigma_z u} \exp\left[-\tfrac{1}{2}\left(y/\sigma_y\right)^2\right] \exp\left(-t/\tau\right)\, dy\, dx \tag{3-30}$$

where B is the half-width (crosswind) of the area source. For conditions in the near-field where $\sigma_y << B$ Equation (3-30) can be approximated by:

$$\chi(x,0,0) = \int_{0}^{x} \frac{2\bar{Q}}{\sqrt{2\pi}\,\sigma_z u} \exp(-t/\tau)\, dx \tag{3-31}$$

Substituting $\sigma_z = ck^d$ and integrating we depart from the original analysis (Lucas 1958) and obtain.

$$\chi(x,0,0) = \frac{2\bar{Q}\; x^{1-d}}{\sqrt{2\pi}\; uc(1-d)} \exp(-t/\tau) \tag{3-32}$$

Because of the approximation used, this result is appropriate for an area source as large or larger than the watershed size (characteristic length scale $\sim$ 0-10 km, of interest in the multi-media model. Also, either the watershed is directly downwind of the area source or is within three characteristic length scales away from the result in Equation (3-32) over $0 < X < S$ to obtain:

$$\bar{\chi} = \frac{1}{S}\int_{0}^{S} \chi\, dx = \frac{2\bar{Q}\, S^{1-d}}{\sqrt{2\pi}\; uc(1-d)(2-d)} \exp(-t/\tau) \tag{3-33}$$

which corresponds to the original Lucas result.

For downwind distances at which $\sigma_z >$ .15B, we can revert to the point source formulas by matching σ_y to the half width of the area source plume thereby determining the downwind distance from a virtual point source. Let us suppose that σ_y is 3S id approximated by S (i.e., assuming no lateral spread of the area source plume for three diameters downwind). Then X_v^y, the distance from the virtual point source which results in the same lateral mixing is given by:

$$x_v^y \sim (S/a)^{1/b} . \qquad (3\text{-}34)$$

The appropriate virtual source location would be different in order to match the vertical mixing observed in the city, however, and is given by $X_v^z = (\sigma_{z0}/c)^{1/d}$ where σ_{z0} represents the vertical mixing observed within the city.

For the far field of the area source, the point source formulas with H=0 are used with σ_y and σ_z calculated as functions of distance from the virtual point source; i.e.:

$$\chi(x,y,0) = \frac{\bar{Q}S^2}{2\sigma_y\sigma_z u} \exp\left[-\frac{1}{2}\left(y/\sigma_y\right)^2\right] \exp\left(-t/\tau\right) \qquad (3\text{-}35)$$

for $x < x_L$

with a log X interpolation on log x between X_L and $2X_L$.

For receptors which are within an urban area, the Hanna Gifford operation used in the TEM model is applied. This is described in the appendix.

3.1.5 Chemical Decay Term

The concentrations of toxic materials will generally be small compared with those of background or emitted organic material in the atmosphere. For this reason, the photo-oxidative breakdown of the toxic material will not be a significant contributor to the free radical pool in the atmosphere. These radical concentrations will be controlled by sunlight intensities, major organic pollutant concentrations, and oxides of nitrogen concentrations. The attack rate of the active intermediate species upon the toxic contaminant molecules will be determined by the radical concentrations and the toxic chemical concentrations. The free radical levels will be assumed to be determined by measurements or models independently. Because of negligible feedback of the

TYPING GUIDE SHEET

toxic substance on the total photo-chemical activity, the radical concentrations can be assigned values appropriate to urban or non-urban conditions. Then the pseudo-first order rate constants for each radical species are represented as the products of second-order rate constants and the independently determined radical concentrations.

Each concentration formula has a factor of exp $(-t/\tau)$ applied to reflect the possibility of chemical degradation. For organic pollutants in air the dominant mechanisms for degradation are direct photolysis, attach by ozone (O_3) attack by hydroxyl radicals (OH) and attack by oxy- or peroxy-radicals (RO_x). The pseudo first order decay expression for X is formed from four terms as follows:

$$-\frac{dx}{dt} = \left(k_{ph} + k_{OH} + k_{O_3} + k_{RO_x}\right) x \qquad (3\text{-}36)$$

The k's are the rate of constants for the removal of the pollutant and examples of actual values have been tabulated (Aravamudan, et al., 1979). Thus, the value of τ is simply the reciprocal of their sum:

$$\tau = \left(k_{ph} + k_{OH} + k_{O_3} + k_{RO_x}\right)^{-1} \qquad (3\text{-}37)$$

The value of time in each expression is calculated from:

$$t = x/u \qquad (3\text{-}38)$$

where x is downwind distance from a point source. For cases within the area source, the wind speed is constant, the emission rate is averaged over the entire area, and the chemistry is linear; therefore an average time for all of the emitting elements is:

$$t = x/2u \qquad (3\text{-}39)$$

where is distance travelled from the upwind edge of the area. Similarly, for locations beyond the near field plume:

$$t = \frac{x - s/2}{u} \qquad x > S \qquad (3\text{-}40)$$

EPA-287 (Cin.)
(4-76)

TYPING GUIDE SHEET

3.1.6 WET DEPOSITION

Wet deposition rates are estimated using the scavenging ratio approach presented by Gatz (1976). That approach has been validated for several toxic organics by Tucker & Preston (1982). The scavenging ratio is simply the ratio of rainfall concentrations to ground level air concentrations. Theoretically, the scavenging ratio would be expected to fluctuate depending on the process by which the contaminant is incorporated in the raindrop. Various processes which have been studied include the condensation of water onto fine particulate in clouds (nucleation scavenging), other in-cloud scavenging processes (rainout), impaction of falling raindrops with particles below the cloud base (washout), and reversible or irreversible absorption of gases. Of these mechanisms, only reversible absorption of gases would be expected to result in a constant scavenging ratio, which should be the inverse of the Henry's law constant. However, Gatz (1976) has shown that scavenging ratios for particulate contaminants do not fluctuate widely, regardless of meteorological conditions, while Tucker & Preston (1981) demonstrate that scavenging ratios may be estimated within an order of magnitude for a wide range of toxic organics. Consequently, the scavenging ratio approach, with an assumption that the scavenging ratio depends on the physical and chemical properties of the contaminant, appears to be an appropriate way to estimate wet deposition for ALWAS. The effect of plume depletion by wet deposition is approximated as a pseudo-first order loss mechanism. That is, DiDOT defines the pseudo-first order loss rate by wet deposition, k_{wet}:

$$k_{wet} = \frac{rR}{h}$$

where r = scavenging ratio

R = rainfall rate

h = vertical spread of pollutants (inversion height far downwind, $2\sigma_z$ near the source)

k_{wet} is the equivalent to the scavenging coefficient as defined in other contexts (e.g., the ATM model; Culkowski and Patterson, 1976). k_{wet} is simply added to the other pseudo-first order loss rates of eq. 3-37.

EPA-287 (Cin.)
(4-76)

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

3.2 NPSDEP

Modifications of the NPS model (Donigian and Crawford, 1976) have been implemented to facilitate its incorporation into ALWAS. NPS simulates the migration of pollutants in overland flow by use of user-input potency factors which are simply the ratio of pollutant concentration to sediment concentration. The deposition rate (accumulation rate) of pollutants on the land surface is also a user-input. Both the deposition rate and potency factors may be input at monthly resolution. In order to fully utilize the detailed deposition information available from DiDOT, the inputs of these parameters have been expanded in NPSDEP. The deposition (accumulation) rate may now be input at daily frequency. However, potency factors are still input at monthly frequency to conserve in data handling.

In NPS the potency factors were strictly a function of land use with separate potency factors for pervious and impervious areas within each land use. For impervious areas, the only sediment available for transport results from atmospheric deposition since erosion of the impervious surface does not occur. Thus, the potency factor used by NPS appropriately describes the "quality" of the deposited sediment. However, for impervious areas, sediment available for transport is comprised of fines detached from the native soil as well as deposited sediment. the use of a single potency factor for the resulting sediment carried in overland flow implies that native soil and deposited sediment have similar concentrations (potency) of the pollutant simulated. This is clearly not the general case, nor does it represent the situation for which ALWAS is designed, namely to evaluate the effects of deposited contaminants in watersheds where other sources of the contaminant may or may not be present. This problem is resolved in NPSDEP by accounting for the separate contribution of eroded sediment and deposited sediment to the total sediment reservoir available for transport, and each contributing component has a unique potency factor. The potency factor for deposited material is calculated by the DiDOT postprocessor.

The other significant modification to NPS found in NPSDEP is the added capability to simulate the effect of contaminated rainfall. The sediment and contaminant deposited by rainfall are simply passed through the watershed by overland transport without modification. The approach is consistent with NPS' fundamental assumption that pollutant migrates with sediment, and the conclusion of Section 4.4 that, for most toxic pollutants, precipitation scavenging is primarily associated with aerosol phases. Contaminants deposited during snowfall are treated as dry deposition, i.e., they are added to the reservoir of sediment available for transport. This is actually accomplished within DiDOT which, unfortunately, is not as sophisticated as NPSDEP in determining the occurrence of snow. This shortcoming may lead to inconsistent treatment of contaminants deposited via snowfall.

BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"

PAGE NUMBER
EPA-287 (Cin.)
(4-76)

TYPING GUIDE SHEET
BEGIN FIRST LINE OF TEXT HERE
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE
9-1/8"

3.3 Dynamic Version of EXAMS

In linking the EXAMS surface water quality model with NPSDEP and DiDOT, the EXAMS aquatic ecosystem model, AETOX1, was modified. EXAMS is designed to provide two types of containment concentration solution estimates. The first type of solution is a steady-state concentration level for a steady load of contaminants. The second type of solution is a transient solution describing the ecosystem self-purification response assuming that the contaminant loading is completely eliminated with the initial condition being the steady state concentration.

In ALWAS, the actual loadings of contaminants on the water bodies of interest will vary over time. During the simulation period contaminant loading levels can vary; and furthermore, the resultant water quality levels will vary significantly. The types of solutions presently available in EXAMS are not amenable to displaying those effects of time variable loadings.

For these reasons we have made modifications to EXAMS to allow full dynamic operation EXAMS and solution for a one-year time period.

EXAMS must interface with the output from NPS, which have been averaged over daily increments prior to the EXAMS interface. Thus, the input loadings from NPS might appear as shown in Figure 3-2.

In order to obtain the type of response indicated in Figure 3-3, the EXAMS code was modified to calculate the time-dependent response to time-dependent loadings. The time-dependent loadings are averaged over a user-specified interval using an interfacing program which, operationally, interfaces NPSDEP output with EXAMS inputs.

3.4 INTERFACING NPSDEP AND EXAMS

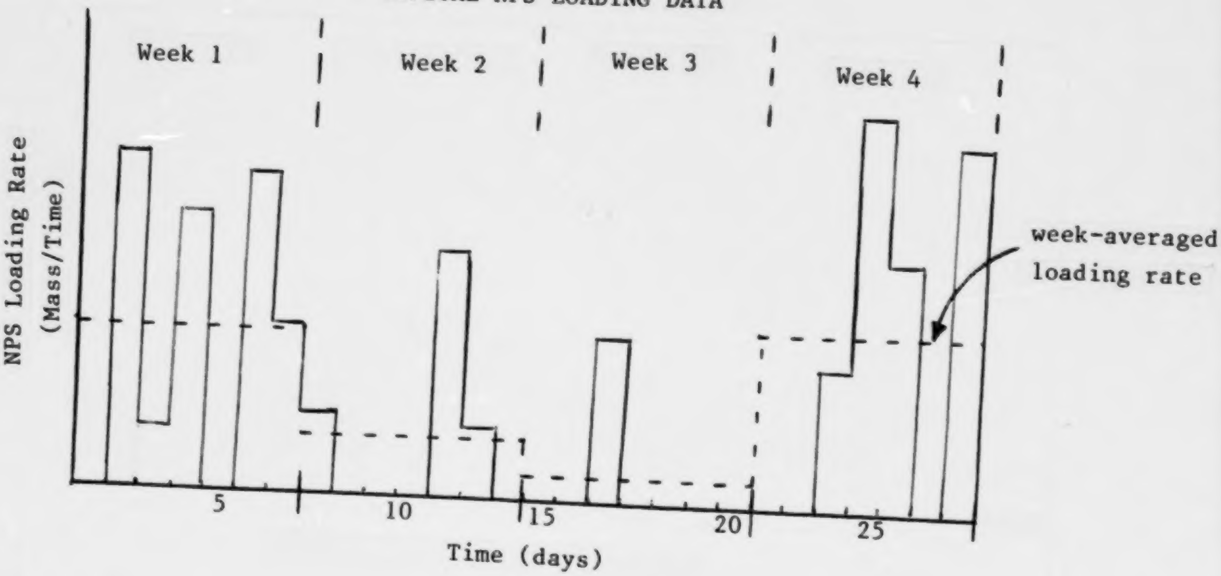
The interface between NPS and EXAMS is a simple transfer of surface runoff and associated sediment and toxicant from the watershed to the waterbody. Numerous "real-world" complexities of the watershed/waterbody interaction are not simulated. For example, interflow and associated toxicant flux are not accounted for. NPS does not simulate contaminant migration via subsurface pathways. Hydrologic coupling of NPS and EXAMS interflow/stream recharge data elements, thus achieving a time dependent interflow input to EXAMS, would have a negligible effect on the final EXAMS results. The authors concluded that addition of such a coupling was not advisable in light of the additional data handling costs to the user.

The domain of an ALWAS simulation may be defined in two quite distinct ways:

- all upland areas contributing runoff to a specific discharge point
- a lowland portion of the watershed, e.g., a flood plain city

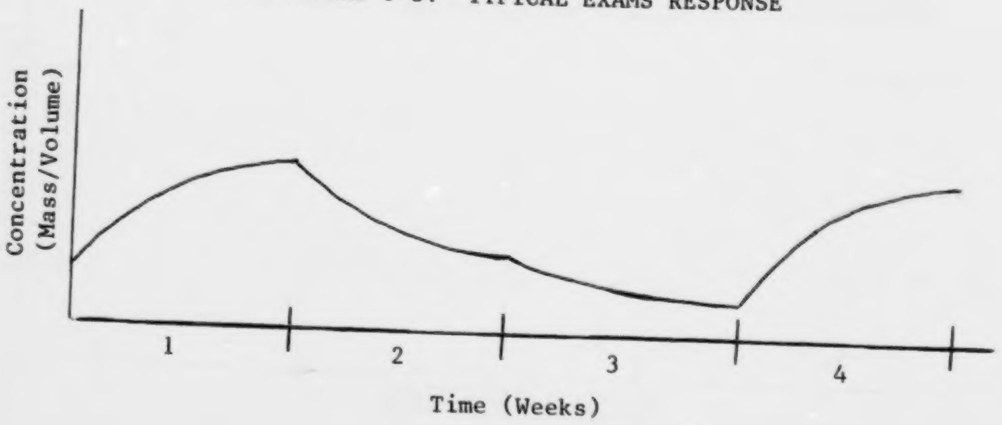
BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

FIGURE 3-2
TYPICAL NPS LOADING DATA



As displayed in Figure 3-2, over a one-week time period, the contaminant loadings from NPS to the stream will vary significantly - particularly as storm events occur (storm events are depicted by the "spikes" in contaminant loading). The actual loading to EXAMS would be an average over a user-specified time period longer than that used by NPS. The "dashed" line in Figure 1 represents the average loading that would be used to represent a one-week loading for EXAMS. The time intervals, approximately the contaminant total half life in EXAMS, can vary depending upon the nature of the contaminant, the water body, temperature, etc. If the time interval were roughly one week, the response of EXAMS will appear as follows in Figure 3-3.

FIGURE 3-3: TYPICAL EXAMS RESPONSE



The existing interfacing logic is well-suited to the first domain type. When evaluating domains of the latter type, the user should note that upstream flows, sediment and toxicant loadings to the EXAMS waterbody (as well as depth, cross-section, etc.) are constant during the simulation period. In reality, these parameters may be expected to vary, exhibiting some degree of correlation with the NPS derived runoff and non-point source loadings. There are no inherent constraints of technical feasibility which prohibit the modification of ALWAS to allow EXAMS to accept time dependent upstream flows and loadings. For that matter, one could allow the whole description of the EXAMS water body to be time dependent (roughly 30 variables required for each EXAMS cell, including total suspended sediment, depth, cross-section, volume, pH, dissolved organic carbon, and so on).

In lieu of practical validation experience with the model which would demonstrate that time dependence was important for certain variables, but not for others, the authors decided to avoid encumbering the user with excessive data collection, manipulation, and handling costs, establishing the only time dependent inputs to EXAMS as those that were delivered by the forerunning software (DiDOT and NPS). Thus some caution is advised when using ALWAS to simulate the lower portion of a larger watershed. Water quality results should be assumed to be valid only during periods (comparable to or greater than the toxicant half life) when the flow, and other waterbody characteristics, match the EXAMS input values. Given the structure of the dynamic version of EXAMS a simple program modification would facilitate time dependent specification of other EXAMS data. This is recommended if the user (a) had data available to support such specificity, and (b) was willing to absorb the additional data handling costs.

TYPING GUIDE SHEET
BEGIN FIRST LINE OF TEXT HERE
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE
BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD, BEGIN SECTIONS HERE

4.0 USER'S GUIDE FOR ALWAS

4.1 GETTING STARTED

4.1.1 BASIC DEFINITIONS AND CONCEPTS

It is difficult to guess just what skills and training a potential user of ALWAS is likely to possess -- will he/she be an air quality specialist looking to apply that training to water quality problems, or a water quality specialist, first learning the jargon of air quality modeling? Whether air or water specialist, use of this model will probably require him/her to become conversant in some new terms. Moreover, we have found it convenient to define familiar terms in ways that have specific, possibly non-standard, definitions when used in ALWAS. The purpose of the section is to familiarize the user with ALWAS jargon.

Portions of the land surface are defined differently in DiDOT and NPSDEP. The basic land unit in DiDOT is the unit square which is a square ranging in side length from 0.5 to 5.0 km. Contaminant air concentrations and air-to-surface fluxes are calculated at the center of each unit square. The domain of an NPSDEP simulation is a watershed. A watershed is all land area upgradient of a specific runoff discharge point. An NPSDEP watershed is a relatively small area (probably less than 5 km^2) for which NPSDEP input parameters have been calibrated by comparison of output with runoff flow and quality. A calibrated watershed must be a small upland area because (see Donigian and Crawford, 1978) NPSDEP does not describe in-channel processes, and only the discharge from a small upland watershed will be unaffected by channel processes. By joint application of NPSDEP with a stream quality model describing sediment transport proceses, NPSDEP plus the stream model may be calibrated against water quality data from a larger basin. Theoretically, ALWAS may be calibrated in this way, but it is not clear, based on present experience, that such an approach would be effective for establishing NPSDEP parameter values.

An EXAMS waterbody may be a stream, river, pond, reservoir, or lake. The EXAMS water body may be subdivided into more than one cell. Cells may represent separately the epilimnion and hypolimnion of a lake, or may represent river reaches (longitudinally connected segments of a river). A segment is a subportion of a basin which contributes runoff to a single cell.

These definitions are illustrated by Figure 4-1, which describes a promising ALWAS usage mode. The river basin depicted has an area of approximately 180 km^2. The area of the NPSDEP calibrated watershed is roughly 4 km^2. The EXAMS water body has been subdivided into four reaches. The reaches indicated define a simple network of streams in the basin, Reaches 1, 2, and 3 representing the main river, and Reach 4 a tributary stream. Networks of this kind are possible in ALWAS, but the complexity which could be represented is limited by the maximum

BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA, OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"

EPA-287 (Cin.)
(4-76)

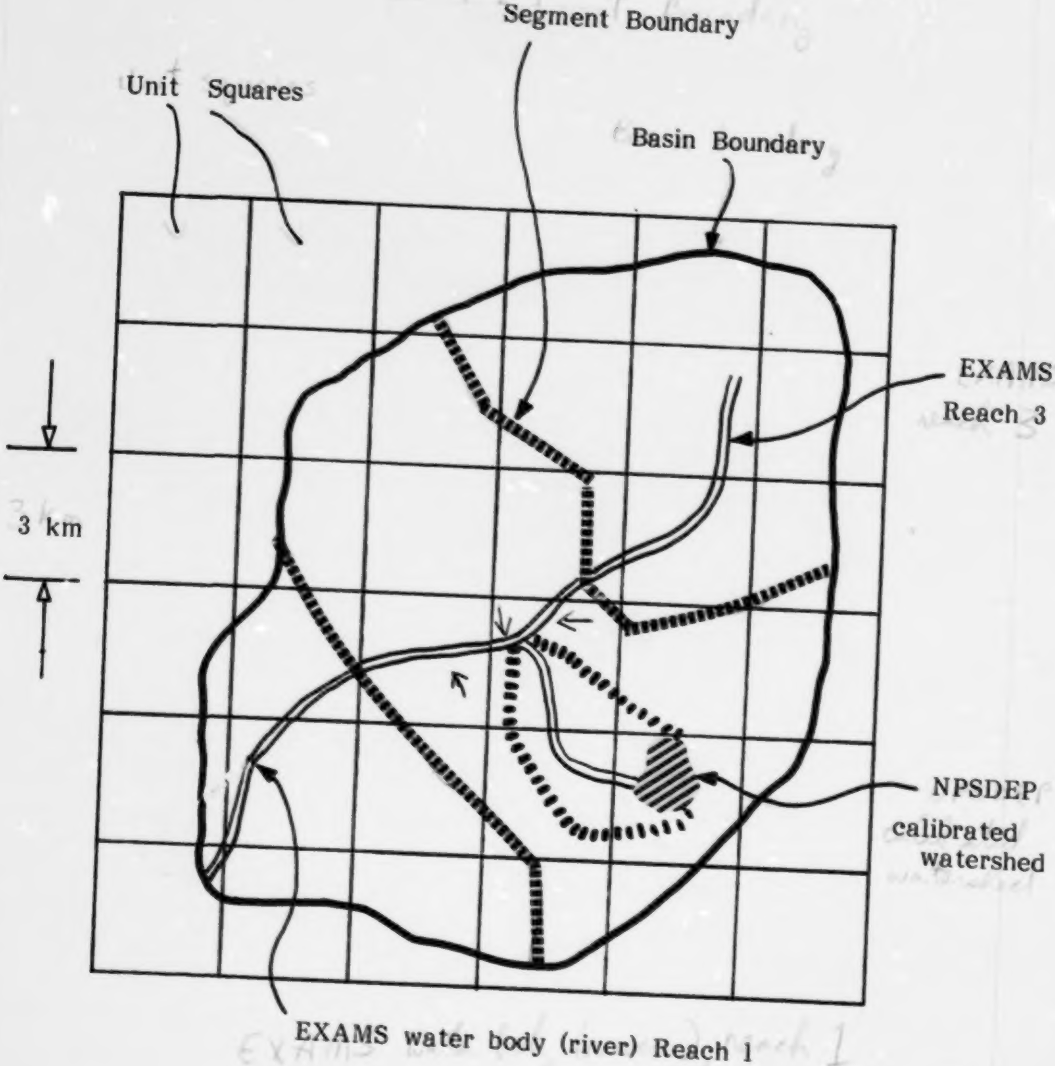

Segment Boundary
Unit Squares
Basin Boundary
EXAMS Reach 3
3 km
NPSDEP calibrated watershed
EXAMS water body (river) Reach 1

FIGURE 4-1. RELATIONSHIP BETWEEN LAND AREA DEFINITIONS USED IN ALWAS

Figure 4-1 Relationship between land area definitions used in ALWAS

2

number of EXAMS cells (reaches in a riverine simulation), which is 10. Each reach is associated with a segment bounded by dashed lines. Non-point source loadings of sediment and toxicant to each EXAMS cell (river reach) would be determined by running NPSDEP using input parameters determined by calibration to the small upland watershed. The only inputs which would vary for simulation of the separate segments would be the DiDOT deposition rates. DiDOT,and its postprocessor, summarize deposition fluxes, calculated at unit square resolution, to yield a segment-average deposition rate. The final step which leads to segment non-point source loadings to separate EXAMS cells is to multiply loadings output from NPSDEP by the ratio of segment area to calibrated watershed area. This "scale-up" is performed by the NPSDEP/ EXAMS interfacing program.

This approach to simulation of river basins is based on two fundamental assumptions:

(1) that the NPSDEP calibrated watershed is representative of the whole basin (except that it may have different deposition rates)

(2) that each small watershed ($5\ km^2$) is linked to the EXAMS water body by a stream channel in which sediment and toxicant transport during storms is rapid and conservative.

"Scale-up" is not necessary to simulate a small urban watershed like the Third Fork Creek watershed in Durham, N.C., which is simulated in the sensitivity analysis test cases described in Section 5. For such a small basin, the NPSDEP calibrated watershed and basin are equivalent. It is also likely, but not mandatory, that in this ALWAS usage mode only one segment would be defined, and it too would be equivalent to the watershed and basin.

4.1.2 DATA SOURCES AND PREPARATION

In order for ALWAS to successfully simulate the transport and fate of a toxicant through the air, soil and water, its various input data requirements must be met. How the user carries this out is discussed in the following paragraphs. This discussion is organized in the same sequence in which ALWAS operates: first the atmospheric, and then soil and surface waters components of ALWAS are presented.

4.1.2.1 DiDOT

Atmospheric emissions are transported, transformed and deposited to the surface by DiDOT in accordance with an input time series of hourly meteorological data. Standard data resources and data processing procedures for preparing the meteorological data for input to DiDOT have been adopted. In particular, the procedure used for application of EPA's current most-sophisticated air quality models was incorporated. The user is referred to the User's Guide for the Single-Source (CESTER) Model for full details of the CRSTER meteorological data preprocessor (CRSMET), but a synopsis of that program and its input requirements is provided below.

The DiDOT preprocessor program is executed independently and prepares meteorological data on a single, full-calendar-year basis in the input format required

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

by DiDOT. The basic function of this program is to transform surface and upper air meteorological information obtained from the National Climatic Center (NCC). The operations performed by this program include:

- Calculate hourly values for atmospheric stability from meteorological surface observations.
- Interpolate twice daily mixing height data to hourly values.
- Reformat other meteorological data to be compatible with DiDOT input requirements.

It is important to note that precipitation data is not treated by the preprocessor at all.

The input data requirements for the preprocessor program consist of two categories of meteorological data - hourly surface observations and twice daily upper air observations. Hourly surface observation data used are:

- Wind direction
- Wind speed
- Dry bulb temperature
- Sky cover
- Cloud ceiling height

These data can be obtained by request from the NCC by specifying a magnetic tape of hourly surface observations in card deck 144 format (CD144) for the specific year of analysis and surface observation stations.

Daily upper air observation data input to the preprocessor are:

- Early morning mixing height
- Afternoon mixing height

The mixing heights are based on NWS upper air soundings at 1200 GMT and 0000 GMT, respectively. These data also can be obtained by request from the NCC in either magnetic tape or tabular form by specifying "twice daily mixing heights" for the specific year of analysis and upper air observation stations. This is referred to as NCC tabulation TD-9689.

The surface and upper air stations should be selected for their meteorological representativeness of the general area being modeled. Generally this corresponds to the stations closest to the emission sources being modeled and in the same climatological regime (e.g., coastal, mountainous, plains). Meteorological data from 1964 are routinely available, although data from any year can be obtained as long as surface data for each hour are available. The year 1964 is convenient because it is the last and most recent year for which routinely

BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
EPA-287 (Cin.)
(4-76)

43

TYPING GUIDE SHEET

reported surface observations are transcribed by NWS on an hourly basis for data analysis purposes. Although NWS stations still report observations on an hourly basis on WBAN Form A, only every third hour is keypunched for computer entry and this is not consistent with DIDOT input requirements. *Thus, there is a* significant increase in the cost of obtaining a magnetic tape of hourly surface observations from the NCC for the years for which hourly data is not already keypunched.

The output of the preprocessor consists of hourly values of wind speed, flow vector, randomized flow vector, stability class, mixing heights and ambient temperature. The flow vector is numerically the 180° complement of the meteorological wind direction.

The specific data requirements of the preprocessor are these:

(a) Surface Station Number. This number identifies the National Weather Service (NWS) surface observation station for which hourly meteorological data is input to the preprocessor. This number is the WBAN (NWS) station identification number, not the WMO block identification number which sometimes is used by the NWS. The Survery of TD-1440 publication tabulates WBAN station numbers for all surface stations for which hourly meteorological data are available from NCC. The surface station selected for the analysis should be representative of the site. The station number is a five-digit integer, input in card form.

(b) Year of Surface Data. This input parameter identifies the year during which the meteorological data was observed, hereafter referred to as the "year of record". Only the last two digits of the year are used, with the input in card form.

(c) Latitude of Surface Station. This input parameter gives the geographic latitude coordinate for the surface station in degrees (north of the equator). The directory of NWS Offices and Stations lists the latitude for each station. The input is in card form.

(d) Longitude of Surface Station. This input gives the geographic longitude coordinate for the surface station in degrees (west of Greenwich). The directory of NWS Offices and Stations lists the longitude for each station. The input is in card form.

(e) Time Zone. This input parameter is a code for the time zone in which the surface station is located. The standard time zone for a given station can be determined from the nationwide map given in any telephone directory. The standard values for this parameter are: 05 = Eastern, 06 = Central, 07 = Mountain, 08 = Pacific. The value corresponds to the number of time zones west of Greenwich Mean Time. The input is in card form.

(f) Number of Days in the Year. This input variable gives the number of days in the year of record. The value input must be 365, or 366 for a leap year. The input is in card form.

EPA-287 (Cin.)
(4-76)

TYPING GUIDE SHEET

(g) Hourly Surface Observations. These data are the "Hourly Surface Observations in the Card Deck 144 Format" available from NCC for the year of record. The preprocessor performs checks for missing data from this file, but can only sense when an entire record (one hour of data) is missing. Thus, if data for only one parameter is absent, the preprocessor will not flag such gaps in the input data. It is recommended, therefore, that the user perform independent integrity and completeness checks. The input of this data is in magnetic tape form.

(h) Mixing Height Data. These data input the nocturnal (minimum) and afternoon (maximum) mixing height for each day in the year of record. The information required for each day includes:

- Upper Air Station Number - this number is the WBAN station identification number identifying the NWS upper air observation station used to calculate mixing heights. The List of Upper Air Stations tabulates such WBAN numbers. The station must be representative of the plant site.
- Year - the last two digits of the year of record for the mixing height data.
- Month - the month number corresponding to a given set of mixing heights.
- Day - the calendar day number corresponding to a given set of mixing heights.
- Nocturnal Mixing Height - the minimum mixing height for a given day calculated from the 1200 GMT upper air sounding on that day.
- Afternoon Mixing Height - the maximum mixing height for a given day calculated from the 1200 GMT upper air sounding on that day.

The input of mixing height data is in card form where one card is required for each day of the year of record. The first card of this file must contain mixing height data for December 31 of the preceding year, while the last card of the file must contain data for January 1 of the year succeeding the year of record. The twice daily mixing height cards must be punched from information available from NCC, on either magnetic tape or in printed tabular form, for an appropriate NWS upper air observation station in the year of record.

(i) Random Digit File: The hourly surface weather observations maintained by the NCC report wind direction only to the nearest 10° of compass heading. This degree of resolution has proved inadequate in air-quality studies, and the standard air-quality modeling practice to overcome this limitation is to vary the nominal 10° wind direction by a random number in the range of -5° to $+4^\circ$ about the nominal value. To insure that variations in the sequence of random numbers used by the preprocessor from one application to another is not in itself a source of error or sensitivity in the final results, the DiDOT

45

preprocessor utilizes on EPA-provided data set of random digits. This is a card-image file of 366 records, each record containing one 24-digit-long random digit sequence. This is used by the preprocessor to calculate the randomized flow vector for each hour of each day. (Note: This adaptation can be easily revoked if a random-number generating sequence is available on the computer system being used; Examination of the FORTRAN preprocessor program supplied to AERL will make the necessary changes obvious).

Execution of the DiDOT meteorological preprocessor will yield two types of output:

(1) an unformatted data set of hourly meteorological data for the year of record, and (2) printed output of program diagnostic messages. The output data set serves as direct input to DiDOT as the hourly values of wind speed, flow vector, randomized flow vector, stability and temperature are structured in the format required by the model. The printed output informs the user of any inconsistencies or missing records in the output tape as a result of deficiencies in the hourly surface observation data input to the preprocessor.

There is one other important type of meteorological data used by DiDOT, namely, the precipitation (rainfall) files. Up to 5 distinct rainfall data sets can be linked directly to DiDOT, to reflect the possibility that more than one precipitation-measuring device (i.e., rain-gauge) may exist in a given watershed or group of contiguous catchments. DiDOT has been structured to use precipitation data in exactly the same format as the NPS hourly precipitation data format, and so the data sources and preparation steps described in the NPS Standard documentation apply also to DiDOT. Precipitation data can be obtained from the National Climatic Center at hourly resolution in their TD-9657 data archive, but this format is not directly compatible with either DiDOT or NPS (NPSDEP). Minor reformatting is necessary and a computer program to do this is easily written. The measurement units of precipitation for both DiDOT and NPSDEP are assumed to be hundredths of an inch.

Pollutant-specific data is required by DiDOT. This includes data on both the characteristics of atmospheric inputs of the two toxicants (one of which should be suspended particulate matter), as well as properties of each toxicant in the ambient atmosphere. Specific parameters in the second category are: atmospheric half-life, settling velocity, deposition velocity, scavenging ratio, and ambient background concentration (due to un-inventoried emission sources). At the present time, current environmental science and research publications and journals are the best resources for most of this information. Estimation methods have been developed, and are presented in Section 4.4.

With respect to suspended particulate matter, EPA maintains a large data archive of ambient particulate matter measurements. The data base is known as SAROAD, and contains air quality measurements from monitoring sites across the nation. Monitoring locations are indexed according to the nature of their exposure (urban, suburban, rural, etc.). Thus, by consulting with the staff of the National Air Data Branch of the EPA in Research Triangle Park, or careful examination of routine publications of NADB, it should be possible to identify an appropriate rural monitoring location which is not unduly impacted by the emission sources explicitly inventoried and input to DiDOT on the basis of

TYPING GUIDE SHEET

distance and direction from the majority of the inventoried sources. For use as an actual value for ambient background, the annual geometric mean concentration or 50 percentile 24-hour average concentration of that monitoring location's set of measurements may be chosen. Local expertise, such as the state air pollution control agency, could also be consulted for guidance and expertise in estimating the background concentration due to un-inventoried emission sources.

Required data on atmospheric inputs (i.e., emissions) of the toxicants being simulated takes the form of an emissions inventory. There are three types of emission sources which can be inventoried and input to DiDOT: point sources, area sources, and city sources. The National Air Data Branch of EPA also maintains a nationwide inventory of emissions of five criteria air pollutants (particulate matter, sulfur dioxide, nitrogen dioxide, hydrocarbons and carbon monoxide), called NEDS. In this data archive, emission rates, locations, and emission source characteristics are available for all significant point sources in any given geographical area. If available, a state-developed emission inventory, such as might have been complied for State Implementation Plan purposes, would be a useful data resource. Similarly, NEDS data files, or State Implementation Plan technical support documents can provide guidance and/or a starting point for the development of area source and city-source inventories for DiDOT.

The estimation of emission rates for non-criteria atmospheric emissions must, in most cases, be done by indirect means. NEDS emission inventories provide information of types and quantities of fuels consumed by combustion processes, so that emissions of combustion-related toxicants (such as BaP) can be estimated. Alternatively, point source emissions of nitrogen dioxide (and sulfur dioxide to a lesser extent) are almost exclusively due to combustion processes and can provide a basis for the scaling of regional BaP emissions. inventoried emissions of hydrocarbons are not speciated in NEDS. Hydrocarbon emission inventories compiled by local air pollution control agencies may be speciated somewhat as a result of control program strategy evaluations, but these are not commonly available. Research reports and Control Technology Guideline documents of EPA may have process-specific emission rates for non-criteria atmospheric pollutants. Risk assessments are another source of general emissions information for specific hydrocarbon pollutants (such as perchloroethylene).

The locations of point sources are coded explicitly on the input to DiDOT. For area sources and city sources, emission squares are specified in a consistent grid system with the square or rectangular grid of receptors (locations at which concentration and deposition are to be calculated). The receptor grid is determined by five parameters: A spacing parameter, the X- and Y-coordinates of the southwest corner of the receptor grid, and two integers representing the number of receptor squares extending to the north and east respectively from the southwest corner. The first three parameters also specify the grid system upon which area source and city source emission squares must be allocated. Namely, the side-length of every emission square must be an integer multiple of the spacing parameter, and the specified coordinates of the southwest corner of every emission square must also be an integral number of receptor spacings from the southwest corner of the receptor grid (it is not necessary, however, for the

EPA-287 (Cin.)
(4-76)

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD: BEGIN SECTIONS HERE

receptor grid to actually extend to each emission source square: in fact, city-sources must be at least as distant from the receptor grid as their own side-length). These geometrical restrictions are consistent with most other current air quality models and can be easily satisfied by careful choice of the receptor spacing parameter and coordinates of the receptor grid corner.

It is assumed that topographic and/or hydrographic/hydrogeological maps of the study area are available to the user, and that a consistent gridding system in kilometers exists or can be devised. On these maps, catchment boundaries should be drawn. Point sources and "upwind" city sources should be identified, and the geographical extent of the city sources, area source grid and receptor grid should be established which provides sufficient coverage of the watershed being studied. The exact location of the southwest corner of the receptor grid should be chosen and then a spacing parameter and grid dimensions should be selected which do not cause an excessively large number of receptor squares to be generated (program maximum is 225 receptor squares). In air quality studies using algorithms similar to those in DiDOT, spacing parameters of 0.5 to 5.0 km are frequently specified. The grid system is then drawn or overlaid on the catchment-boundaries map. After assigning arbitrary identification numbers between 1 and 20 to each catchment and cell, the user can then associate each receptor square with from 1 to 3 catchment identifiers. The problem of receptor squares lying wholly outside any defined catchment or cell, i.e., at the edges of the receptor grid or mapped catchments region, can be handled by assigning one index value, such as 19, as a "buffer region" or "dummy index" to be used for each such receptor square. All results calculated for this catchment identifier will later be discarded by the DiDOT postprocessor. From the mapping of catchment boundaries, locating the point sources, specifying mapping of the five receptor grid parameters, and the overlaying of the grid system on the catchments map, all the geography related inputs to DiDOT can be determined.

4.1.2.2 NPSDEP

The modified version of NPS, named NPSDEP, is designed to use a data set prepared by the DiDOT postprocessor. There are no new data requirements for NPSDEP with the exception of interfacing control cards, including the catchment number.

4.1.2.3 EXAMS

In the ALWAS context, EXAMS is run in batch, rather than interactive mode. Thus, a number of EXAMS interactive user inputs must be specified in a file which must be assigned to Logical Unit Number 5 (SYSIN). The information required is similar to that required by EXAMS (Burns, et al, 1980) with the following additions:

NPRINT - The number of interaction of the Gunge-Kutta integration routine per printout

NPS - Logical flag, TRUE if input prepared by NPS EXAMS

DOT - Logical flag, TRUE if input prepared by DiDOT post processor

BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA, OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"

EPA-287 (Cin.)
(4-76)

PERIOD - Load averaging time (hrs): NPS loadings will be presented to EXAMS as daily totals but will be processed only after time averaging over PERIOD hours

and the following subtractions:

DRFLDG - drift loadings

NPSLDG - non-point source loadings

PCPLDG - rainfall loadings

These latter three variables are now supplied by NPS EXAMS and the DiDOT post processor.

4.1.3 SYSTEM DESCRIPTION

The complete processing scheme comprising ALWAS is shown in Figure 4-2. This diagram indicates logical unit numbers associated with the various input and output files, and the auxiliary and/or optional data processing paths available in ALWAS.

As shown in the diagram, DiDOT is designed to use the output data set of the meteorological data preprocessor principally. This data set is preferred since it minimizes I/O operations. However, DiDOT will accept as an alternative input for hourly meteorological data, card formatted data following the last input package (i.e., after the $ENDATA card). Programs have been provided to carry out this transformation step in either direction.

The unit numbers shown in parenthesis in the diagram indicate that these are nominal unit numbers: the actual numbers that must be specified for these data sets will depend on parameters used by the individual processing program. For example, the numbers assigned to output data sets from the DiDOT postprocessor depends on the spedific numbers between 1 and 20 assigned to each catchment and water body in the DiDOT input, and the number of NPS catchments and EXAMS cells which are included in the run of DiDOT. Full details are provided in Section 4.6.

Symbols shown in Figure 4-2 by dashed lines are optional inputs. Optional card-image hourly meteorological data inputs to DiDOT were mentioned above. The dashed card-deck input to the DiDOT postprocessor indicates that the user can optionally input a baseline potency factors matrix, if the effects of intermittent sources are to be examined. This is described in more detail in Sections 4.2.1.3 and 4.3.1.3.

The divided hexagonal symbol shown with the NPSDEP component represents an automated data preparation step that is required to transfer data sets between the PDP 11/70 at AERL and the IBM installation at NCC. Specifically, the HASP spooling protocol will only receive and transmit logical records of 80-byte length. NPSDEP was designed to read and write logical records of 353-byte length. The program BRKUP is executed to break up the long records into 80-byte records, and RESTORE would be used to reconstruct the long records before NPSDEP

Fig. 4-2

D,DoT component

FIGURE 4-2 ALWAS FLOW CHART: DIDOT COMPONENT

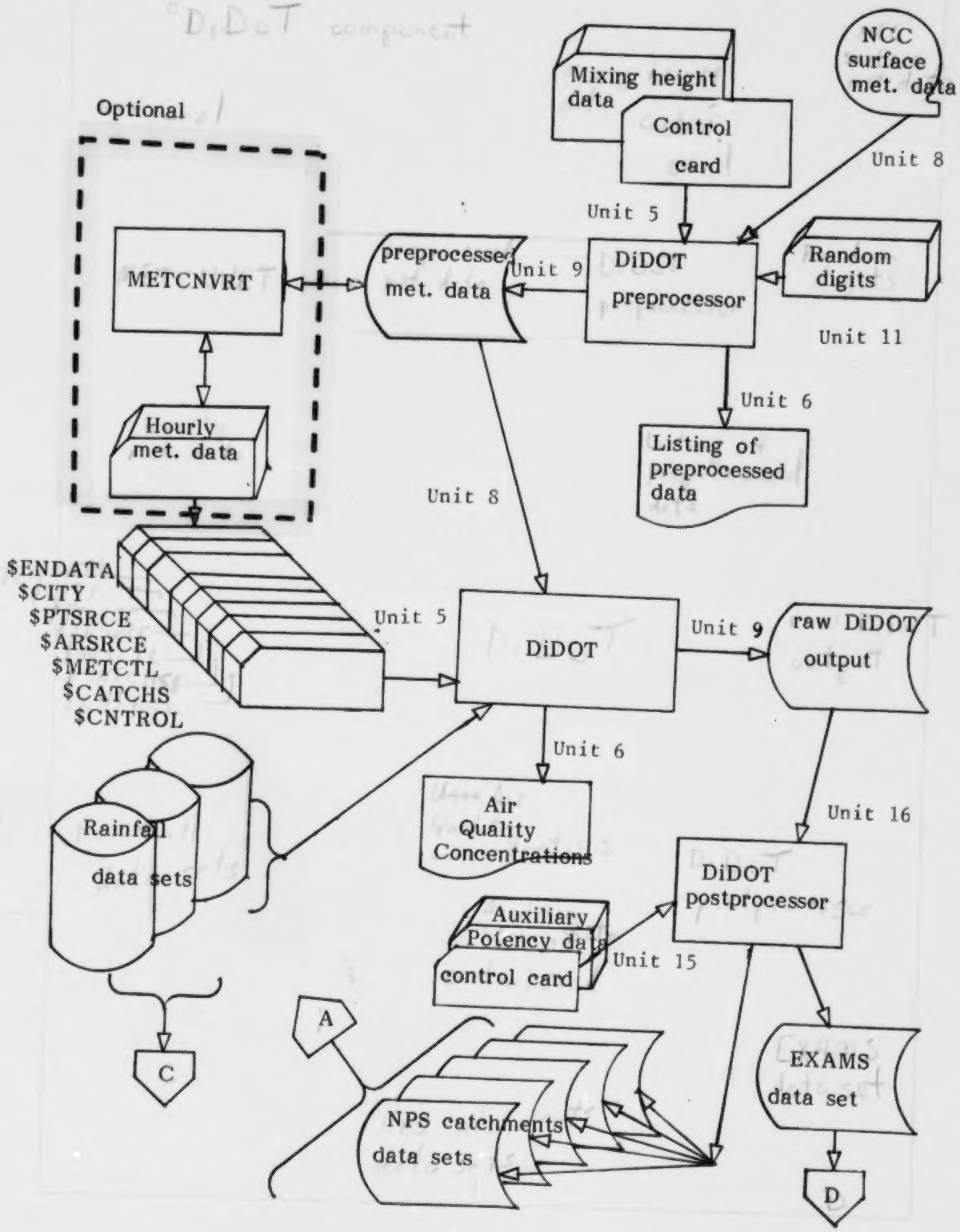

Mixing height data
Control card
NCC surface met. data
Unit 8
Unit 5
Optional
METCNVRT
preprocessed met. data
Unit 9
DiDOT preprocessor
Random digits
Unit 11
Unit 6
Listing of preprocessed data
Hourly met. data
Unit 8
$ENDATA
$CITY
$PTSRCE
$ARSRCE
$METCTL
$CATCHS
$CNTROL
Unit 5
DiDOT
Unit 9
raw DiDOT output
Unit 6
Air Quality Concentrations
Unit 16
Rainfall data sets
DiDOT postprocessor
Auxiliary Potency data
control card
Unit 15
C
A
EXAMS data set
NPS catchments data sets
D

FIGURE 4-2 (continued)

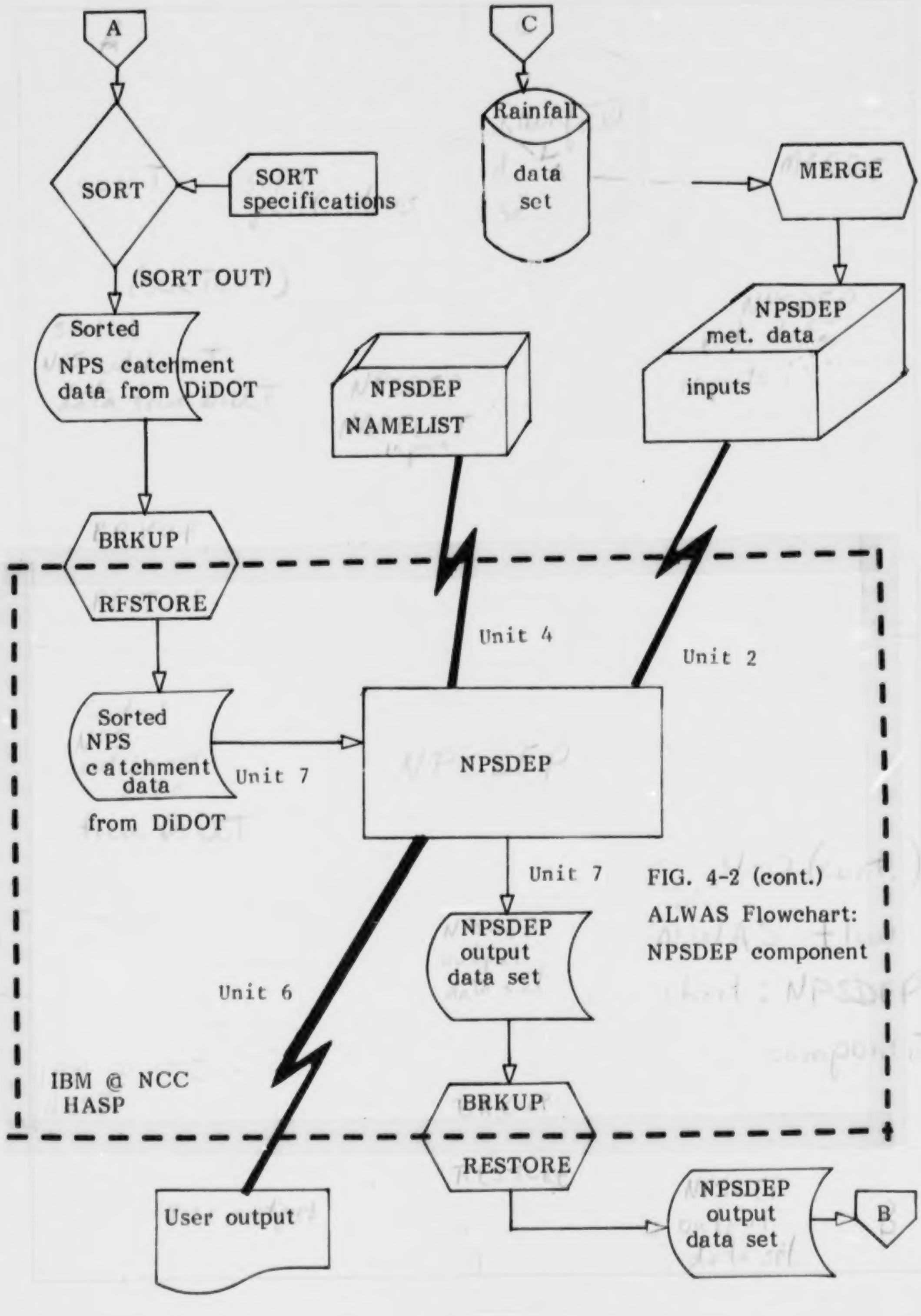

A
C
SORT
SORT specifications
Rainfall data set
MERGE
(SORT OUT)
Sorted NPS catchment data from DiDOT
NPSDEP NAMELIST
NPSDEP met. data inputs
BRKUP
RFSTORE
Unit 4
Unit 2
Sorted NPS catchment data from DiDOT
Unit 7
NPSDEP
Unit 7
NPSDEP output data set
Unit 6
IBM @ NCC HASP
BRKUP
RESTORE
User output
NPSDEP output data set
B

FIG. 4-2 (cont.)
ALWAS Flowchart:
NPSDEP component

FIGURE 4-2 (continued)

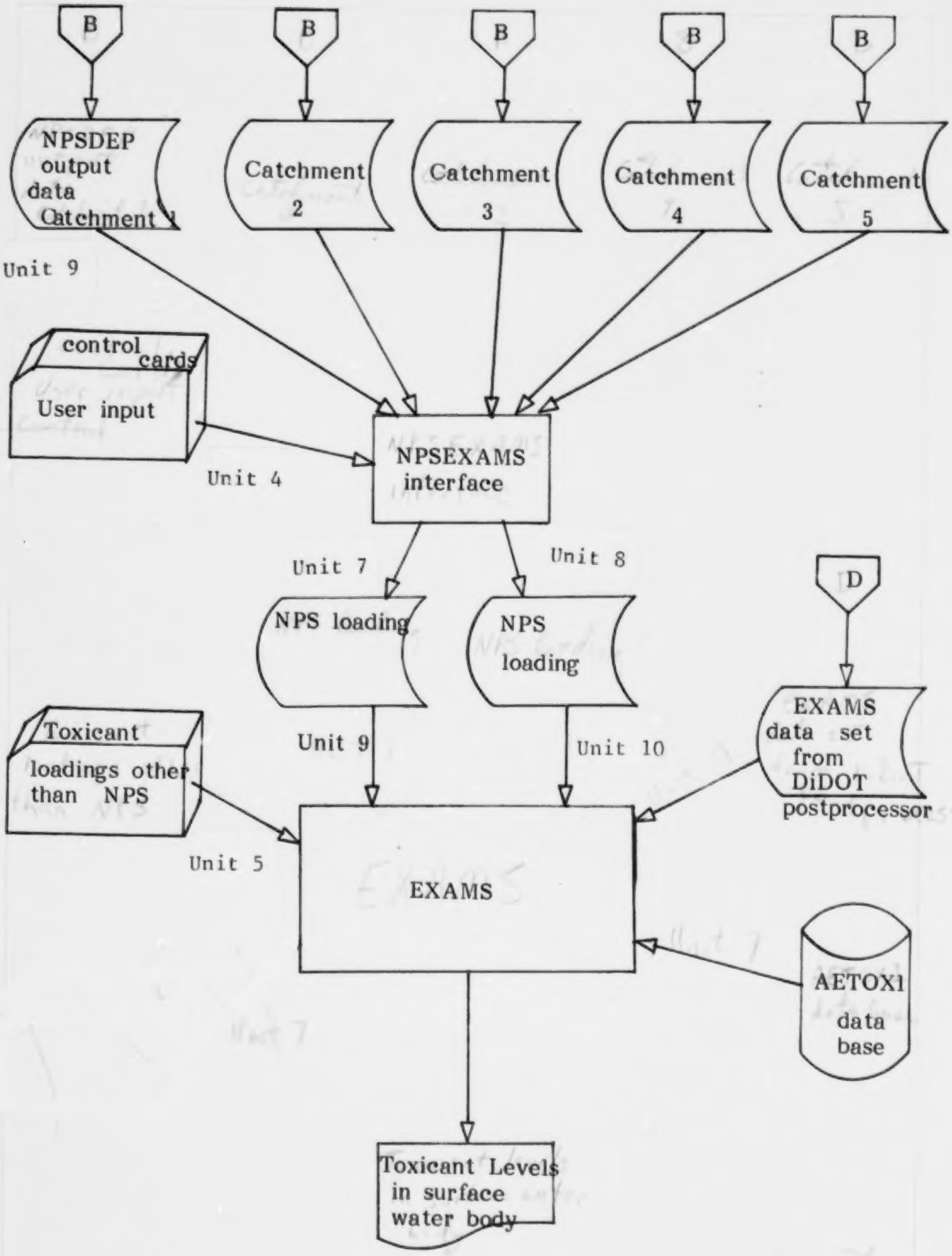

B
B
B
B
B
NPSDEP output data Catchment 1
Catchment 2
Catchment 3
Catchment 4
Catchment 5
Unit 9
control cards
User input
Unit 4
NPSEXAMS interface
Unit 7
Unit 8
NPS loading
NPS loading
D
EXAMS data set from DiDOT postprocessor
Toxicant loadings other than NPS
Unit 9
Unit 10
Unit 5
EXAMS
AETOX1 data base
Toxicant Levels in surface water body

FIGURE 4-2 (cont.) ALWAS Flow chart, EXAMS Component is executed on the IBM main frame and before further processing is done on the NPSDEP output data set on the 11/70.

It is important to note that NPSDEP treats one catchment per run, whereas EXAMS treats an entire watershed. Therefore, the output files for each catchment must be concentrated (appended) together into a single input file for EXAMS. As shown in the diagram, this operation is carried out before the NPS to EXAMS interfacing program is applied.

4.2 PROGRAM DESCRIPTION

4.2.1 DiDOT MODEL COMPONENT

4.2.1.1 PREPROCESSOR

The meteorological data preprocessor is a small FORTRAN program whose purpose is to integrate two distinct types of routine meteorological parameters into a single data set for use by DiDOT, and to minimize the I/O burden on DiDOT in reading this data.

The preprocessor carries out the following operations:

(a) Air temperatures are converted from Fahrenheit to Kelvin

(b) Wind speeds are converted from miles/hour to meters/second. Wind speeds reported as 0 miles/hour (i.e., calm winds) are raised to a fixed minimum value of 1.0 m/sec.

(c) Wind directions to 10° compass resolution are converted to flow vectors (i.e., the supplementary angle of the nominal wind direction) and randomized within -5 and +4 degrees of the nominal flow vector. If winds are calm (direction and speed coded as 00,00), the wind direction of the last non-calm hour is used and randomized.

(d) The hourly stability class (Pasquill-Gifford-Turner classification scheme) is determined from the time of day, the amount of cloud cover, and the wind speed. The table used for doing this was given in Section 3.1.

(e) Hourly values of the mixing height are determined according to each of the two interpolation schemes, using the input values of each day's morning minimum and afternoon maximum mixing height as fixed points. The two interpolation algorithms differ on the basis of the study area being more typified by urban or rural micrometeorology. The version of the preprocessor provided retains the standard feature of providing both an urban and a rural mixing height for each hour, but DiDOT uses only the rural mixing height values. The interpolation algorithm requires the exact time of sunrise and sunset for each day of the year at the meteorological observing station. These are determined by the program from the input values for station latitude, longitude, and time zone, and equations for computing the solar angle.

73

(f) The computed values for each parameter for each hour are accumulated in 24-element vectors. At the end of each day, a single unformatted record of 684 bytes is written out, and the same data is formatted and printed.

The user is referred to documentation of any of the current sophisticated air-quality models (RAM, ISC, CRSTER) for more information on the meteorological data preprocessor.

4.2.1.2 DiDOT

The DiDOT program itself is composed of a relatively small main program and 38 subroutines. The logic flow of the program is such that overlaying is possible to meet memory limitations of the PDP 11/70.

The program executes in three phases: an input and initialization phase, an execution phase, and a wrap-up phase.

The input and initialization phase reads in and processes the input data packages. The routines and their functions which are called during this phase are:

TABSET -	This routine calculates and stores values of certain exponential and other functions in tables in a labelled common block. These functions are called repeatedly during program execution and they are stored in tables because repeated look-ups are presumed to be computationally faster than recomputation. TABSET also turns off the run-time system error monitor control for floating point underflow errors.
METCTL -	METCTL is called in response to the $METCTL input card. It reads and prints the input data associated with this input package. This routine determines whether preprocessed meteorological data is available to the program or logical unit 8.
CATCHS -	This routine processes the data in the $CATCHS input package. It in turn calls the routines XYLOC and INDXCK to assist in constructing the receptor vs. catchment and catchment vs. precipitation file relationships. Pertinent information is written on the unformatted data set about NPS catchment identifiers and EXAMS identifiers also.
PTSRCE, ARSRCE, and CITY -	These three routines are called in response to the corresponding package name cards. Each one checks independently that the source number sequencing is consecutive. ARSRCE and CITY also check that the southwest corner location and side length of each source square is consistent with the receptor grid system. The routine examines the six values of the initial sigma-z (SZ0) for each city-source, and supplies non-zero default values if values are input.

The PTSRCE routine computes and stores certain parameters relating to the effective stack height. Namely, in two arrays dimensioned as (3, 10, 10), PTSRCE determines the final effective stack height and the downwind distance at which this is reached for each stack and at nominal wind speed values and 3 stability regimes. The equations of Briggs are used to calculate H_p and X_{Hf} under unstable through neutral conditions, slightly stable, and extremely stable conditions. The ten nominal wind speeds range from 0.67 m/s through about 12 m/s. These computed results are stored in arrays which are declared in a labelled common block.

KPOLLS - This routine is called after the $ENDATA card is encountered. It summarized total emissions of each pollutant for each source category and ensures that there are non-zero emissions for both pollutants.

It should be recalled that the main program processes the input associated with the $CNTROL input package.

During the execution phase of program execution, several "utility routines" are called repeatedly and from various other routines. Before discussing the main routines of the execution phase, these utility routines are described.

VD - This routine carries out the hour-by-hour adjustment, if necessary, of the user-input pollutant-specific deposition velocity, on the basis of the dynamic meteorological variables of stability class and wind speed.

FLUX - This routine computes the deposition rate at a given receptor as a function of the local air concentration and the (corrected) deposition velocity.

HOURLY - This subprogram extracts the hourly meteorological parameters into unsubscripted variables from vector arrays. It also converts wind direction data from degrees into radians and calls VD to calculate adjusted deposition velocity for the hour for each pollutant.

INDEXER - As the program executes, concentration and deposition flux results are directly obtained at each receptor location in the gridded area. Each receptor is associated with from one to three catchments, and it is necessary to allocate or assign results at the level to the catchments with which it is associated. The routine provides the relationship between receptor and catchments.

DUMMY - This routine is called simply to insure that the proper overlay is loaded at the proper point in program execution.

UPDATE - This routine is used to maintain the tables of highest hourly concentration contribution at any receptor from any one of each of the three source categories. Thus, there are effectively 6 tables that are updated: one table for each source type and pollutant. UPDATE also controls the computation of the deposition flux for each source-receptor-hour.

TYPING GUIDE SHEET
BEGIN FIRST LINE OF TEXT HERE
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

XLCOMP, - SYCOMP, SZCOMP, EXCOMP	This group of functions is used to provide values on X_L, σ_y, and exp $(-1/2_p{}^2)$, respectively. This is done in each routine by first examining the magnitude of the input arguments against predetermined maximum and minimum values, and based on that determination, returning the function value either by look-up and interpolation in tables generated by the TABSET routine, or explicit calculation of the function.

The remaining routines comprising DiDOT can best be explained in the context of the logical flow through the program. Fundamentally, DiDOT is driven by the array IDAYS, provided through the $METCTL input packages which specifies the days which are to be simulated. The main program (DiDOTPGM) calls routine GETMET which increments a day number index, and compares it to the value stored for this day in the IDAYS array. The main program calls GETMET repeatedly until a flag returned by GETMET is non-zero. If this flag is negative, the end of the input meteorological data set was encountered, and the program proceeds to the wrap-up phase. Otherwise, a day to be simulated has been found, and GETMET will have obtained and processed the necessary meteorological data for this day. The GETMET routine obtains the hourly meteorological data for the day on the basis of whether the non-zero value in IDAYS is +1 or -1. If positive, an unformatted record is read from Unit 8; if negative, 24 card records are read on Unit 5 (standard card input). In either case, values for the year, month, and day of the month are read from the input record(s).

Based on the NPPT parameter, the routine seeks hourly precipitation in the specified data sets. NPPT specifies the number of data sets, and the array LUNPPT gives the logical unit numbers. GETMET synchronizes precipitation data with the meteorological data by matching the values of year, month, and day of month between them. GETMET totals precipitation by data set per day, and total precipitation in all data sets for the day. This latter value is used later in the program to control whether any wet deposition calculations will be performed.

The main program calls routine PRESET for each day that is "turned on". This routine uses the background concentration data and the hourly (adjusted) deposition velocity to establish concentration and (dry) deposition that is due to uninventoried emission sources in the study area.

DiDOT treats point sources, area sources, and city sources in three separate blocks, and these are skipped if there are no sources of that category. Within each block or source category, the concentration and deposition of each source to each receptor is determined explicitly for each hour. Thus, there is a DO-loop over hours of the day repeated in each source-category block of the program, rather than having a single hourly loop extending over the three source categories. This approach was taken to minimize the frequency of overlaying during program execution. Specifically, all sources of a given category are processed for the whole day from a single overlay.

Point sources are handled by the routines PTGEOM, PTCONC, CERMAC, CONCEN, DEPRAT, PLRISE, and HEFFIX. The PTGEOM routine examines the source-receptor-wind angle geometrical relationships for all point sources for each receptor-hour. It determines the downwind and crosswind distances between the sources and

BOTTOM OF IMAGE AREA, OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
BEGIN LAST LINE OF TEXT
3/8"

EPA-287 (Cin.)
(4-76)

5b

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

each receptor and sets flags to specify whether a given receptor is upwind or too far crosswind of a given source, in which case the remaining calculations are skipped, or whether the receptor is closer or further from the source than the precomputed distance to final effective stack height for the indicated wind speed and stability class. The PTCONC routine cycles over each point-source and receptor combination for each hour of the day, subject to the flag set by PTGEOM. If the flag indicates that the source is closer to the receptor than the distance to final effective stack height, the PLRISE routine is called to calculate ΔH and H at the actual distance. The value for effective stack height is then adjusted, if necessary, for stack downwash effects by the HEFFIX routine. The CONCEN routine ultimately calculates the concentration of the point source at the receptor, by calling the CERMAK and DEPRAT routines to apply the Ermak analytical solution and to calculate the net deposition rate respectively. As discussed earlier, these algorithms are invoked by CONCEN on the basis of downwind distance and meteorological (i.e, stability class and mixing height) relationships. CONCEN returns the concentration value to ITCONC, which applies a units conversion factor and the chemical decay term. Finally, for this source-receptor-hour, PTCONC calls UPDATE, to rank the calculated value, compute the deposition, and add the concentration and deposition values into the accumulating arrays PPTCON and DAYDEP.

The concentration due to area sources in calculated by the routines ARGEOM, ARCOMP, and ARCONC. The routine ARGEOM is first called to examine the wind direction for each hour of the day (the pattern of dispersion from individual area sources in only a function of wind direction). The routine ARCONC calculates the geometry and source strength factors for each of the 4 upwind area source grid squares and calls ARCOMP for the computation of the concentration at the specific receptor. It applies the chemical decay term and calls UPDATE.

City sources are treated by the routines CTGEOM and CTCONC. The CTGEOM routine computes downwind and crosswind distances and upwind virtual source distances for each city source. Similar to the PTGEOM routine, CTGEOM examines whether the entire receptor grid is upwind or too far crosswind from all of the city sources, and which concentration algorithms should be applied to select the logic path for the calculation of concentration. It contains a modified version of Ermak expression, the expression derived by Lucas previously presented, and equations for treating dispersion and deposition from virtual point sources, including calls to the DEPRAT routine. CTCONC applies the units conversion factor and the chemical decay term before calling UPDATE.

For each day, the main program loops sequentially through all point sources, area sources, and city sources. At the end of this chain, the routine DAYEND is called. The DAYEND routine performs averaging of concentration and deposition results over catchment areas and time, to meet the specifications and definitions of NPSDEP and EXAMS. The daily average, catchment-average concentration value is sent to the DAYUPD routine, which maintains a ranking table of the ten highest daily concentrations for each pollutant. DAYEND also calls the routine OUTPUT, which prepares and writes both dry and wet deposition data in unformatted mode on the putput data set (Unit 9). This routine uses the data in the PPTCON and DAYDEP arrays, the IRAIN array, and the liquid/frozen precipitation parameter TSNOW to calculate dry and wet deposition fluxes. In

BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"

PAGE NUMBER
EPA-287 (Cin.)
(4-76)

TYPING GUIDE SHEET

the absence of precipitation for a given hour, all deposition is treated as dry deposition. If there was precipitation, the ambient concentration in PPTCON is multiplied by the scavenging ratio for each pollutant and the rainfall amount (converted to metric units) to calculate the wet deposition flux. However, if the ambient temperature during the precipitation event is below TSNOW, the deposition is presumed to be associated with snowfall and added to the dry deposition flux. Otherwise, it is output as part of the wet deposition flux.

Upon return from DAYEND, the main program loops back to the section which calls GETMET. Program execution proceeds in this fashion until the end of the meteorology input is encountered at which time DiDOT calls the routine WRAPUP. This routine prints the tables that have been maintained by the UPDATE and DAYUPD routines. Figure 4-3 shows a simplified flow chart of DiDOT program logic.

4.2.1.3 POSTPROCESSOR

(1) Accepts, as input from DiDOT, catchment definitions (and whether the catchments are NPS- or EXAMS-defined), daily dry flux ($\mu g/m^2$) per pollutant per catchment, and hourly wet flux ($\mu g/m^2$) per pollutant per catchment.

(2) Accepts, as optional user input, monthly background potency factors (tons toxicant/tons TSP) per catchment for two toxicants and five years.

(3) Produces, for NPS, daily dry TSP flux (tons/acre) per NPS catchment, montly average potency factors (tons toxicant/tons TSP) per NPS catchment, and hourly wet flux (tons/acre) per pollutant per NPS catchment.

(4) Produces, for EXAMS, daily dry flux (kg/m^2) per pollutant per EXAMS catchment, and daily wet flux (kg/m^2) per pollutant per EXAMS catchment.

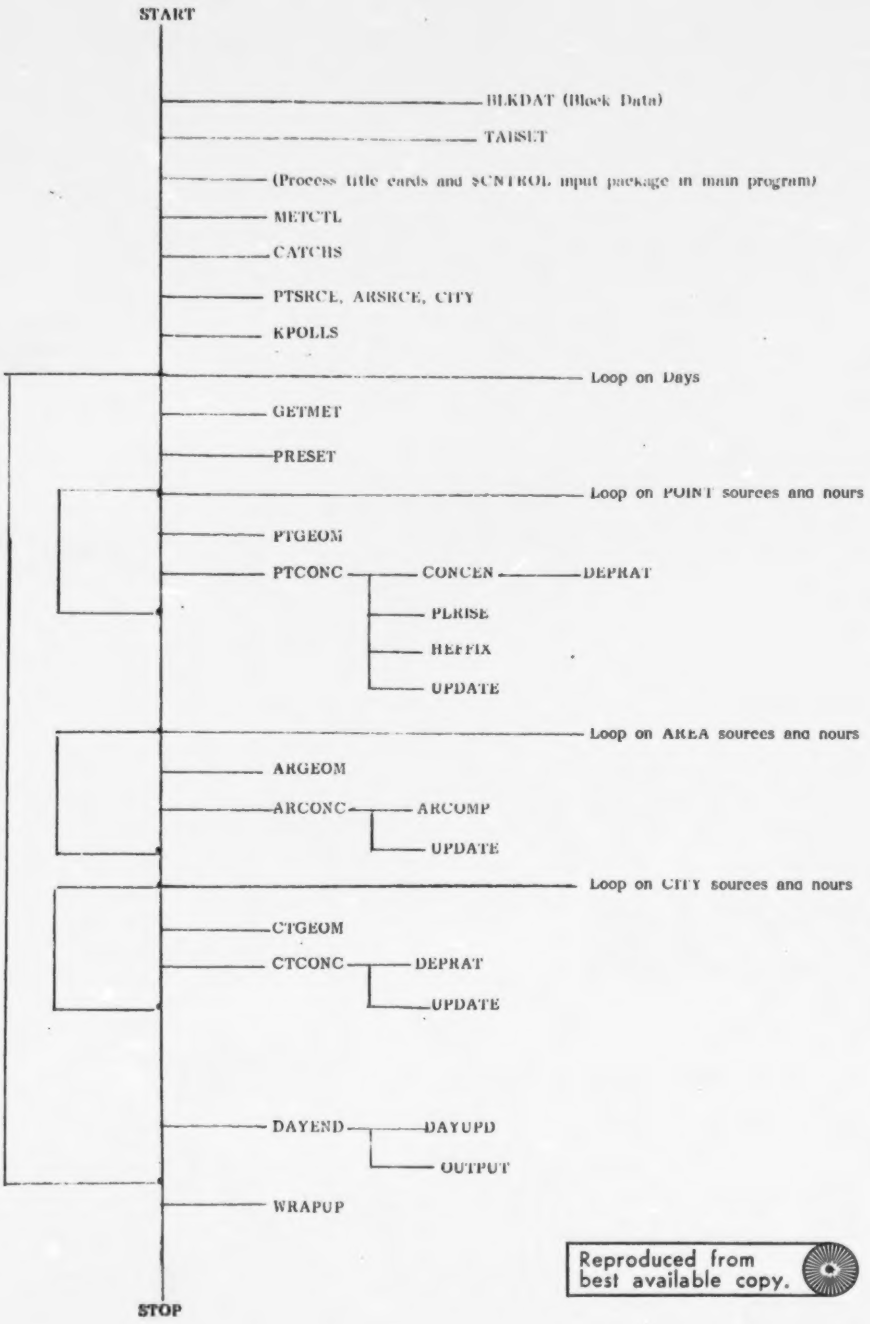
Prepended to each NPS-related output flag is a twelve digit sort flag, so the DiDOT Postprocessor output to NPS may be sorted into the correct order for NPS input.

Routines in the DiDOT Postprocessor are as follows:

Routine	Function
MAIN:	Calls INIT to initialize arrays and flags, then reads a record type and calls REC1, REC2, or REC3, or stops, depending on the record type. When the end of the input file is reached, PRTPOT is called to calculate and print the final monthly potency factor(s), and PRTTSP is called to print the daily TSP flux matrix.
INIT:	Initializes key flag variables, and calls GETPOT to get the optional monthly background potency factors.

EPA-287 (Cin.)
(4-76)

1958


START
BLKDAT (Block Data)
TABSET
(Process title cards and $CNTROL input package in main program)
METCTL
CATCHS
PTSRCE, ARSRCE, CITY
KPOLLS
Loop on Days
GETMET
PRESET
Loop on POINT sources and hours
PTGEOM
PTCONC
CONCEN
DEPRAT
PLRISE
HEFFIX
UPDATE
Loop on AREA sources and hours
ARGEOM
ARCONC
ARCOMP
UPDATE
Loop on CITY sources and hours
CTGEOM
CTCONC
DEPRAT
UPDATE
DAYEND
DAYUPD
OUTPUT
WRAPUP
STOP

Reproduced from
best available copy.

FIGURE 4-3

DiDOT FLOW CHART

GETPOT:	Reads in the optional monthly background potency factors, if these are given.
REC1:	Reads the number of pollutants and the catchment definition vector from a record of type 1, and checks these for correctness.
REC2:	Reads dry flux information from a record of type 2 and processes it. If the month changes, NEWMON is called, and if the year changes, NEWYR is called. Prints daily flux for EXAMS cells.
REC3:	Reads wet flux information from a record of type 3 and processes it. If not enough record type 3's are given in sequence, the program is halted. Prints daily flux for EXAMS cells.
PRTTSP:	Prints daily TSP flux per NPS catchment for one year. Called by NEWYR when the year changes, and by MAIN when the end of the input data file is reached.
PRTPOT:	Calculates and prints monthly potency factors, based on the monthly total flux of each pollutant, and on the monthly background potency factors if they are given. Called by NEWMON or NEWYR when the month or year changes, and by MAIN when the end of the input data file is reached.
DAYMON:	Given the day number (1-366) and year number, DAYMON returns the month and day of the month. Called by REC2 to determine which month a day belongs to.
NUMDAY:	Given the month and year, the NUMDAY function returns the number of days in the month as its function value. Called by PRTPOT, if monthly background potencies are given, for the purpose of calculating weighted average potencies.
NEWMON:	Called by REC2 when the month changes. Calls PRTPOT to calculate and print monthly potency factors, and calls MONCLR to clear the monthly potency factor array.
NEWYR:	Called by REC2 when the year changes. Calls PRTPOT to calculate and print monthly potency factors, calls PRTTSP to print daily TSP flux per catchment for the year, calls MONCLR to clear the monthly potency factor array, and calls YRCLR to clear the TSP flux array.
MONCLR:	Clears the monthly potency factor array.
YRCLR:	Clears the TSP flux array.

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

4.2.2 NPSDEP

NPSDEP is a modified version of the NPS program, which is documented by Donigian and Crawford (1976). The modifications accomplish five primary objectives:

- input parameters and sources (files) were modified to facilitate linkage with DiDOT
- algorithms for handling deposited sediment and toxicant were modified to facilitate linkage with DiDOT - providing for independent simulation of dry deposited contaminant and contaminant arising from land use activities.
- incorporation of a simple algorithm for simulation of contaminated rainfall, or wet deposition.
- a "bug" in the SNOWMELT calculations was corrected to bring the FORTRAN code into conformance with the description of SNOWMELT provided by Donigian and Crawford (1976).
- output parameters and sequencing was modified to facilitate interfacing with EXAMS.

The resulting program, NPSDEP, may be run in conjunction with DiDOT and EXAMS (to form ALWAS) or may be run alone. When run alone it is virtually equivalent to NPS.

The program structure, with respect to subroutine names, functions, and calling sequences is identical to NPS, with one exception: a new subroutine, TDIST, was added which computes the hourly temperature distribution used by SNOWMELT (described further below).

Input Modifications: The user inputs are modified to allow the option of running NPSDEP in conjunction with DiDOT and EXAMS as part of ALWAS or standing alone. If DiDOT has been operated, NPSDEP will read the DiDOT postprocessor output file with daily dry deposition rates and hourly wet deposition rates. If DiDOT has not been operated the input is similar to that required by NPS, with monthly inputs of accumulation rates, but with the added flexibility of unique potency factors for deposited, as opposed to generated, sediment. Details are given in 4.3.2.

Deposited and Generated Sediment Handled Separately: In order to allow distinct patterns of non-point source pollution from deposited airborne contaminants vs. land use activities which might, by distinct processes, contaminate the land surface, the algorithms which account for the reservoir of fine sediment on pervious areas have been modified. In NPSDEP, two reservoirs of sediment are accounted for on pervious areas. SRER is the reservoir of accumulated or deposited sediment, while SGEN is the reservoir of generated or detached fines. The total reservoir of fines on pervious areas is given by:

$$STOT = SRER + SGEN$$

BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA, OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"

PAGE NUMBER
EPA-287 (Cin.)
(4-76)

The fraction of the total reservoir which is deposited is given by:

RFRAC = SRER/STOT

During storms, the sediment washoff, which is equivalent to a waterbody loading, is assumed to come fractionally from the SRER reservoir (RFRAC of the washoff) and the remaining fraction from the SGEN reservoir (1-RFRAC). Since the deposited and generated fractions have different potencies, the potency of the washoff is the weighted average of the two, i.e.:

AVGPOT = PMP*(1-RFRAC) + PMD*(RFRAC)

where PMP and PMD are the potency factors (updated monthly) for the pervious generated sediment and the deposited sediment, respectively. This representation permits unique patterns of deposited and generated toxicant (at a land use resolution) which is characterized by Figure 2-3.

Contaminated Rain: During rainstorms (runoff producing events) the wet deposited sediment is added to the total reservoir of deposited material, SRER and subjected to the same washoff constraints as the total suspended sediment load.

The pollutant deposited with rainfall is simply added to the pollutant washoff for the day, implying that all contaminant deposited with rain within the day is delivered to the stream. This algorithm is not a good description of pollutant deposited with snow, so deposition by snow is treated differently. The difference actually arises in DiDOT rather than NPSDEP. DiDOT independently determines whether a precipitation event is rain or snow. During snowfall the "wet" deposition is simply added to the dry deposition total for the day. Wet deposition is zero during snow. Thus, the pollutant deposited during snow simply stays at the land surface until rain or melt causes runoff to occur.

SNOWMELT: The NPS SNOWMELT algorithm as presented by Donigian and Crawford (1976) is subject to errors in time sequencing and subroutine data transfer. These errors may be machine-dependent, since, to our knowledge, it has only been observed on a UNIVAC computer. These have been corrected in NPSDEP with the introduction of a new subroutine, TDIST, which computes the hourly temperature distribution from the daily minimum and maximum temperatures. TDIST is called from NPSDEP Main Program once per day.

Output Modifications: NPSDEP prepares user output and intermediate output designed for EXAMS linkage, to user defined logical units (files or printer). The user output is similar to NPS output as described by Donigian and Crawford (1976).

The linking of NPS to DiDOT and/or EXAMS is primarily at the job control level, and mainly involves careful manipulation of certain data files. Indeed, with the few exceptions, noted below, a single run of the current revised version of NPS requires no additional input that was not required in the original version.

Insofar as the user is concerned, the only new variables to contend with are two logical flags, DOT and EXAMS, which are specified in BLOCK DATA. If set equal to TRUE, DOT will cause NPS to expect the presence of an additional data file from the DiDOT postprocessor discussed in 4.3.1.2; similarly, if EXAMS = TRUE, NPS will write onto an output file the date and daly totals for runoff flow (m^3), sediment washoff (kg) and toxicant washoff (kg). This output file, after suitable modification, will then serve as input to EXAMS.

The DiDOT postprocessor file (logical unit number 7) will contain for this particular catchment, daily accumulation rates for sediment (tons day^{-1} $acre^{-1}$), monthly potency factors and hourly wet deposition rates for sediment and toxicant (tons day^{-1} $acre^{-1}$) on those days which have rain. The user should note the extreme care which must be taken in managing data files if more than one NPS catchment are to be modeled simultaneously, because each catchment will have unique input and output data sets. This will become clearer from the discussion on interfacing NPS and EXAMS in Section 4.2.4. The user is further reminded that the input precipitation data for this particular NPS catchment must be the same as that used by DiDOT. For data checking purposes the user may set the variable IDBUG = TRUE; among other things, this will cause the echoing of the postprocessor data onto the printed output.

In linking NPS to DiDOT, the user must set the variable MNVAR = 1 to allow for monthly variation in potency. Furthermore, the variable PMPTAB, which specifies potency of generated fines over pervious areas, must be set equal to an appropriate value for the toxicant at hand, as it remains unaffected by DiDOT input. The user may ignore the accumulation variables ACUPV and ACUIV, as well as the potency variables PMDTAB and PMITAB, because they will be replaced by values which come from the air quality model.

When EXAMS = TRUE, NPS will write a single record onto the bulk output file (logical unit number 3) for each day of the study, including days with zero runoff. This file will later be used by the interfacing routine NPSEXAMS in providing input to EXAMS.

The user has the option of linking NPS only to DiDOT or only to EXAMS. This can be easily done.

To link only to EXAMS, the user merely sets DOT = .FALSE., which causes NPS to ignore the presence of any input files from the DiDOT postprocessor. The usual accumulation rates and potency factors from the NPS Namelist input set are then used in calculating the flow, sediment and toxicant runoff. This technique may also be repeated for up to ten catchments, with all NPS output files "collated" in the usual way by the interfacing routine NPSEXAMS (see section 4.2.4).

To link only to DiDOT, the user may proceed in one of two ways:

1) If it is felt that wet deposition is insignificant, the user may set EXAMS = .FALSE., and DOT = .TRUE.; the wet deposition records from the DiDOT postprocessor will then still be read in, but they will not figure into the total runoff as given in the normal NPS printed output. By setting IDBUG = .TRUE., however, the user may still examine the daily wet deposition totals separately, as discussed above.

EPA-287 (Cin.)
(4-76)

2) To see the total daily runoffs including wet deposition, the user may set EXAMS = .TRUE. and simply examine directly the output on file number 3. This may then be compared with other printed output from NPS as file 3 also contains the date for each daily record. In so doing, the user is cautioned to have a conversion table handy, as the output on file 3 will be in cubic meters and kilograms, not cubic feet and tons. To get more legible output, the user may also simply use file 3 as the input to NPSEXAMS, and run NPSEXAMS for the single catchment in debuging mode.

4.2.3 EXAMS - Dynamic Version

The only substantive changes to EXAMS, as documented by Burns, et al., (1980) are in SUBROUTINE's GHOST & FCT. GHOST reads data streams prepared by the DiDOT postprocessor and the NPS/EXAMS interface program. GHOST now iterates its CALL's of the substantive water quality subroutines, CKLOAD, FIRORD, STEADY, FLXOUT, and RKFYS, repeating these CALL's until it encounters an end-of-file on the input loading data file. During the first such iteration, GHOST allows the subroutine STEADY to develop the steady state solution for the first input load. This cell-by-cell concentration vector Y(J) is then input to RKF45 to determine the time dependent solution.

On subsequent iterations the initial concentration vector delivered to the integrator routine, RKF45, is the same as the final concentration vector of the last time step. Modified program is shown in Figure 4-4.

The other significant change to EXAMS is the incorporation of a time dependent load in the time dependent solution, as obtained by RKF45. This change appears in SUBROUTINE FCT, called by RKF45. FCT calculates the time derivatives for the concentration in each cell. FCT was changed so that the derivative line looks like:

```
DY(J) = -Y(J) * TOTKL(J) + CONLDL(J)
```

where Y is the mass of contaminant, TOTKL the total first order loss rate, CONLDL the loading rate, and DY the rate of change of mass, all in cell J.

4.2.4 NPSEXAMS

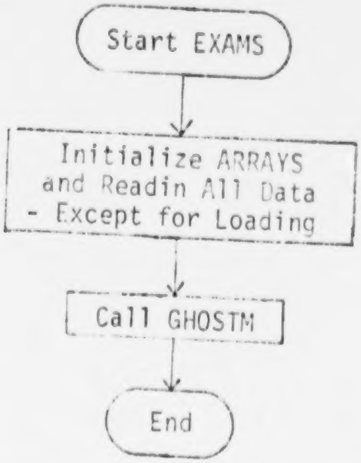
NPSEXAMS transfers surface runoff flow, and associated sediment and toxicant to EXAMS. The interfacing routine between NPS and EXAMS is a small program which acts mainly as a collation of output files from NPS for processing by EXAMS. Key assumption which allows the collation is that runoff from any particular NPS catchment provides loadings to only one EXAMS cell. Furthermore, EXAMS contains no intracell geometry, thus if two different catchments provide loadings to the same EXAMS cell, the effect is additive, i.e., on any given day the EXAMS cell "sees" only a single load which is the sum of the two catchment loads. The summation and sorting for each day are the only calculations done.

This requires, then, a user-supplied mapping which specifies these flow directions. This is contained in the variable MAP, where MAP(I) is defined as the

FIGURE 4-4

EXAMS DYNAMIC VERSION: MODIFIED FLOW CHART

MAIN PROGRAM (Modified)


Start EXAMS
Initialize ARRAYS and Readin All Data - Except for Loading
Call GHOSTM
End

GHOSTM (Modified GHOST)

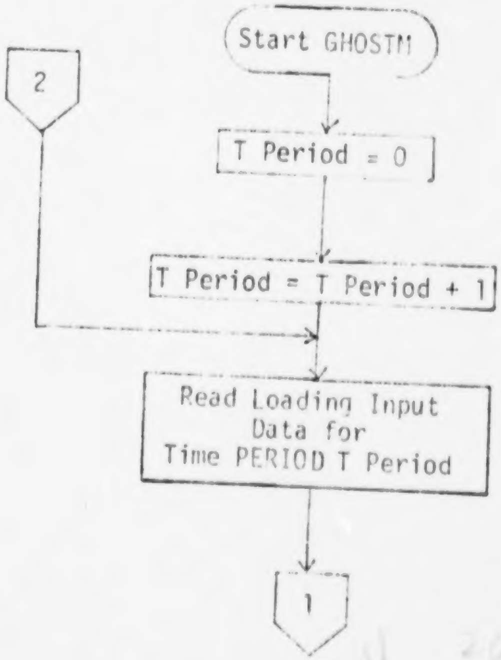
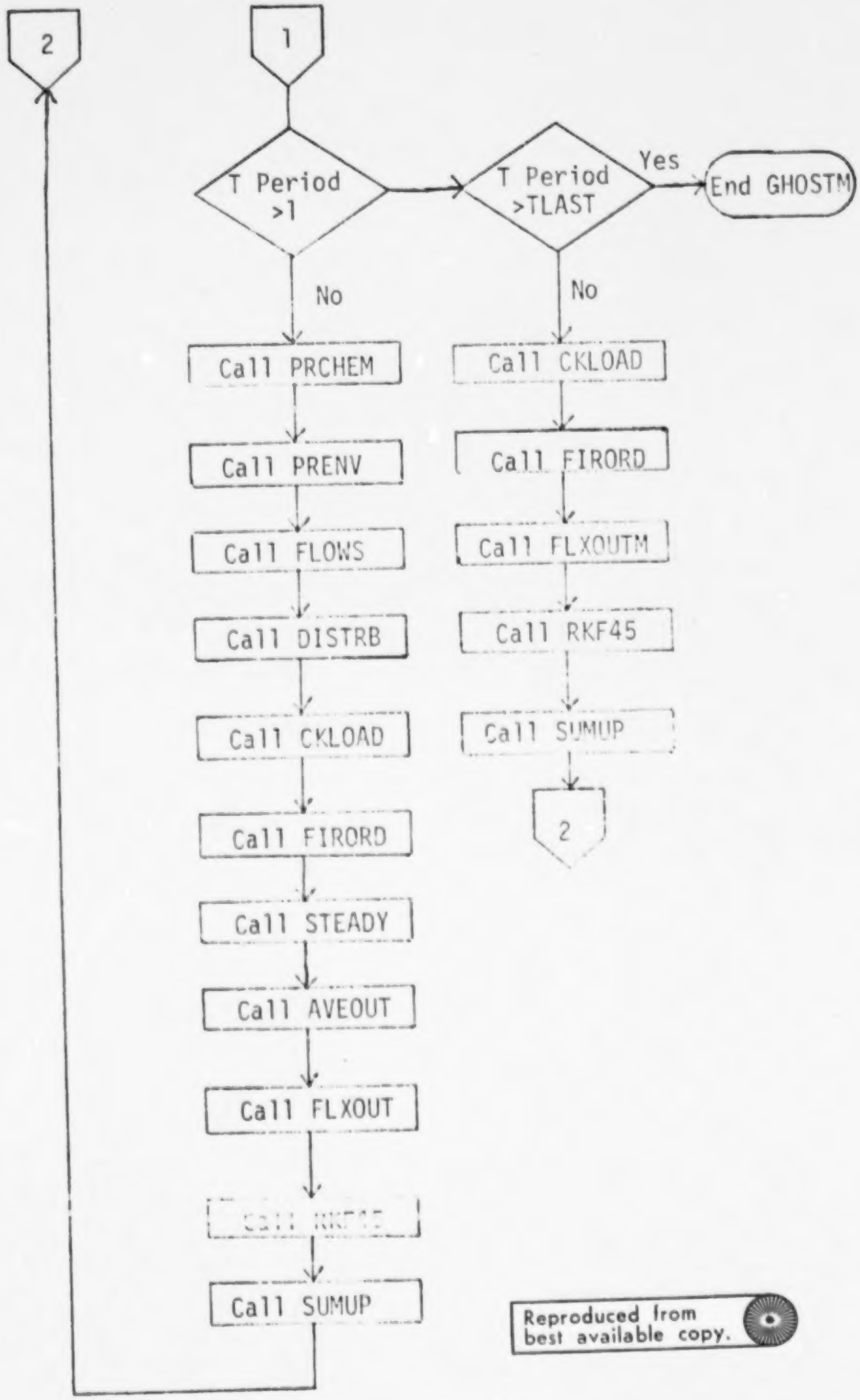

2
Start GHOSTM
T Period = 0
T Period = T Period + 1
Read Loading Input Data for Time PERIOD T Period
1

FIGURE 4-4 (Continued)



2
1
T Period >1
No
Call PRCHEM
Call PRENV
Call FLOWS
Call DISTRB
Call CKLOAD
Call FIRORD
Call STEADY
Call AVEOUT
Call FLXOUT
Call RKF45
Call SUMUP
T Period >TLAST
Yes
End GHOSTM
No
Call CKLOAD
Call FIRORD
Call FLXOUTM
Call RKF45
Call SUMUP
2

Reproduced from best available copy.

27

EXAMS cell (1-10) which receives the loadings from the I^{th} NPS catchment. Currently up to 10 NPS catchments are allowed.

The user also has the option of approximating the runoff from a larger area than that which has been explicitly modeled in an NPS run by simply scaling up the loadings by the ratio of the areas. This could be a valid option, particularly if the NPS input parameters have been finely calibrated to a very small watershed, and if the user has good reason to believe that the smaller watershed is highly representative of the larger catchment of which it is a subset; cf. the previous discussion in 4.1.1 of our differentiation between catchments and watersheds. In any event, the user must specify these catchment/watershed ratios in the variable SCALE, even if they are all unity.

The user must also specify the number of NPS catchments and total number of days in the study.

The need for a sorting routine is bypassed at the job control level, in that the user specifies different logical unit numbers to the output files for the various NPS catchments. In particular, catchment k's output file must be on logical unit number k-8, and MAP and SCALE should reflect this. Otherwise there is no restriction in the numbering scheme so long as it agrees with EXAMS cell definitions.

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

4.3 INPUT DATA FORMATS

4.3.1 DiDOT

4.3.1.1 PREPROCESSOR

The preprocessor requires an initialization card, the random digits file, the daily mixing heights file, and the hourly surface observations file (tape). The first three are described in Tables 4-1, 4-2, and 4-3. The input deck structure is very straightforward. The initialization card occurs first, then the mixing heights cards. It is important to note that the mixing height deck begins with the last day of the preceeding year and ends with the first day of the following year. In practice, this is done by replicating the first day and the last day of the year of the input data, and changing the date fields accordingly.

The format of the hourly surface weather data used by the DiDOT preprocessor is that of the National Weather Service's CE-144 data family. A reference manual including card formats is available from the National Climatic Center, Asheville, N.C.

4.3.1.2 OVERVIEW OF INPUT DATA REQUIREMENTS

4.3.1.2.1 DATA CATEGORIES

There are essentially six data categories required for input to DiDOT. These data are necessary for providing all source data, receptor grid information, control information and specifying requirements for reading the meteorology and rainfall data from other files.

All the source information is provided in three different packages which are denoted by the following names:

1. $PTSRCE;
2. $ARSRCE; and
3. $CITY.

As indicated by their names, $PTSRCE cards provide the point source characteristics, $ARSRCE cards provide the area source grid characteristics and $CITY contains the city source characteristics. The inputs required for each of these packages are discussed in depth below (Detailed Description of Input Data Packages).

The three remaining data categories provide additional information needed to run DiDOT. These package names are:

BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"
PAGE NUMBER

TABLE 4-1

PREPROCESSOR INITIALIZATION CARD FORMAT

Card Columns	Format	Description
1-5	I5	National Weather Service Surface Station identification Number
6-7		Year of Surface Data
8		Blank
9-18	F10.1	Latitude of the Surface Station (degrees to hundredths)
19-28	F10.1	Longitude of the Surface Station (degrees to hundredths)
29-30	F2.0	Time zone in which the Surface Station is located: 05 = Eastern 06 = Central 07 = Mountain 08 = Pacific
31-34	I4	Number of days in the year of record (365 for non-leap years; 366 for leap years)

TABLE 4-2

PREPROCESSOR MIXING HEIGHT DATA CARD FORMAT

Card Column	Format	Description
1-5	I5	NWS Upper Air Station WBAN Number
6-7	I2	Year of record (last two digits)
8-9	I2	Month
10-11	I2	Day
12	1X	Blank
13-17	F5.0	Morning Mixing Height (m)
18-30	13X	Blank
31-35	F5.0	Afternoon Mixing Height (m)

TABLE 4-3

PREPROCESSOR RANDOM DIGITS FILE FORMAT

Card Column	Format	Description
6-8	I3	Julian Day Number (1-366)
12-35	24I1	String of I4 random digits, 0-9, used for wind flow vector randomization

1. $CNTROL;
2. $METCTL; and
3. $CATCHS.

The $CNTROL package contains information pertinent to each pollutant being modelled, such as atmospheric half-lifes, settling and deposition velocities, background concentrations and scavenging ratios. It also contains the roughness length of the terrain being modeled, the spacing of the receptor grid and the coordinates of the southwest corner of the receptor grid. The $METCTL (met control) package contains the information required to read meteorology and rainfall data from their respective files. The days of the year to be simulated are specified in this package. Also specified are the number of rainfall files used by DiDOT and the logical unit numbers they are associated with. It also contains an option for the program to print the meteorology of the days being processed. The threshold temperature for frozen precipitation (snowfall) vs. liquid precipitation (rainfall) is also specified. The $CATCHS package contains the information required to assign receptor grid squares to catchments, and indicate which rainfall files should be used for specific catchments. It also contains the dimensions of the receptor grid and specifies which catchments are considered as EXAMS waterbodies or waterbody cells. The parameters for all these packages are discussed in depth in Section 4.3.1.2.3.

The $ENDATA card is normally the last card in the run stream data and denotes the end of the input data package. It is discussed in Section 4.3.1.2.3.

DiDOT requires meteorology and rainfall data that are not in the input data run-stream. These files are identified by the $METCTL input package and read by the GETMET subroutine. A detailed description of these files can be found in Section 4.3.1.2.3. Detailed Description of Input Data (Other than Card Input Data Stream).

4.3.1.2.2 Input Data Packages

The names of all the data packages were presented in Section 4.3.1.2.1. This section will discuss the format of the package name card, the program's processing of the package name card and the flexibility in structuring the input data stream.

Each package name card, i.e., $CNTROL ...... etc., is 6 or fewer letters long and is preceded by a dollar sign. This dollar sign must occur in the second column of the card for DiDOT to recognize the package name and call the appropriate subroutine. This format is also true for the $ENDATA card. The dollar sign must be in the second column for DiDOT to recognize the end of the input data packages and to start processing. In all cases, the first column on the package name card will be left blank and the second column must have a dollar sign. The package names begin in column 3 and match the spelling and format for card columns 3 to 8.

When DiDOT identifies the package name card, it calls the corresponding subroutine to read in the data. For example, when $METCTL is read, DiDOT calls the subroutine $METCTL that reads the data and stores it for DiDOT's further use. It is extremely important for the package name card to be formatted correctly. The $CNTROL package is processed by the main program.

DiDOT is very flexible in the structuring of the input data stream. Only two requirements must be met for the program to read the data correctly. The $CNTROL package must be the first data group in the input data stream. After that, the packages may be placed in any order considered appropriate by the user. The second requirement is, as mentioned before, the input data stream must end with an $ENDATA card. Keeping these two requirements in mind, the user can structure the input data stream as desired. The $CNTROL, $METCTL, and $CATCHS input packages should each occur only once in the input stream. The program has array capacities for up to 10 point sources, 50 area sources, and 5 city sources. As each source of each type is individually sequenced on its respective package name card (see Tables 2,3, and 4), these bounds are the limits on the number of times each such package can occur in the input stream.

DiDOT provides the user with the capabilities to thoroughly document all pertinent features of a specific simulation. This is done via an unlimited number of title cards, which preceed all other input packages in the input stream. The user can specify any desired geographic, temporal, and pollutant-specific commentary for the particular application on as many cards as are necessary. All 80 columns of the card are read by DiDOT and transferred to the printed output (thus, occurring on the first page of the printed output) until a card with the first 8 columns being all blank is encountered. In that point, DiDOT begins scanning therest of the input data stream for recognizable package name card.

4.3.1.2.3 Detailed Description of Input Data Packages

A. $CNTROL

The $CNTROL package provides data for variables that are constant throughout the modeling period. These variables include the geographic extent of the study area, settling velocities, deposition velocities, scavenging ratios, background concentrations, and half life of each pollutant and roughness length.

The coding of these parameters is explained in detail in Table 4-4.

B. $PTSRCE

The $PTSRCE package reads the point source characteristics into DiDOT. There are three cards per point source; the coding of the data for each point source is shown in detail in Table 4-5.

C. $ARSRCE

The $ARSRCE package reads the area source characteristics into DiDOT. There are two cards per area source; the coding of these cards is explained in detail in Table 4-6.

D. $CITY

The $CITY package reads the city source characteristics into DiDOT. There are three cards per city source; these are coded as described in Table 4-7.

TABLE 4-4

Input Formats for $CNTROL Package

Parameter	Card No.	Card Column	Format	Description
b$CNTROL	1	1-8	A2, A6	Identifies input of global constants; b = leading space.
UNIT	2	11-20	F10.0	Side length of a unit square in receptor grid, km
XSWCOR	2	21-30	F10.0	X coordinate of the SW corner of the receptor grid, km
YSWCOR	2	31-40	F10.0	Y coordinate of the SW corner of the receptor grid, km
RUFLEN	2	41-50	F10.0	Roughness length, m
TAU(1)	3	11-20	F10.0	Half life of pollutant 1, hour^{-1}
TAU(2)	3	21-30	F10.0	Half life of pollutant 2, hour^{-1}
VSETTL(1)	4	11-20	F10.0	Settling velocity of pollutant 1, m/s
VSETTL(2)	4	21-30	F10.0	Settling velocity of pollutant 2, m/s
VDEPOS(1)	5	11-20	F10.0	Deposition velocity for pollutant 1, m/s
VDEPOS(2)	5	21-30	F10.0	Deposition velocity for pollutant 2, m/s
SCAVNG(1)	6	11-20	F10.0	Scavenging ratio of pollutant 1, unitless*
SCAVNG(2)	6	21-30	F10.0	Scavenging ratio of pollutant 2, unitless*

(continued)

Table 4-4 (Continued)

Parameter	Card No.	Card Column	Format	Description
BKGRND(1)	7	11-20	F10.0	Background concentration of pollutant 1, μg/m^3
BKGRND(2)	7	21-30	F10.0	Background concentration of pollutant 2, μg/m^3

* The scavenging ratio expresses the relationship of the concentration of a pollutant in air to its concentration in precipitation. Thus, its units are implicitly (μg/m^3 air)(μg/m^3 precipitation). For reference, 2 hundredths of an inch of rain falling on an acre of land yields over 2 cubic meters of rain.

Table 4-5

Input Data Formats for $PTSRCE Package

Parameter	Card No.	Card Column	Format	Description
b$PTSRCE##	1	1-10	A2, A6, I2	Identifies input of point source data on next two cards. The #'s indicate the number of the point source, i.e., $PTSRCE01, $PTSRCE02, ... etc. These numbers must be encountered sequentially in the input stream. b = leading space.
NAMES	2	3-10	2A4	Point source name, i.e., boiler stack
Q1	2	11-20	F10.0	Emission rate (g/s) of pollutant 1
Q2	2	21-30	F10.0	Emission rate (g/s) of pollutant 2
X	2	41-50	F10.0	X coordinate of point source (km)
Y	2	51-60	F10.0	Y coordinate of point source (km)
HEIGHT	3	11-20	F10.0	Physical stack height (m)
DIAM	3	21-30	F10.0	Inside diameter (m) of stack
VELOC	3	31-40	F10.0	Stack exit velocity (m/s)
TEMPK	3	41-50	F10.0	Stack exit temperature (°K)
VOLFLX	3	51-60	F10.0	Volume flux of pollutants (m^3/s); not needed if one has DIAM and VELOC data coded

Note: Point source package (3 cards each) can be repeated for up to 10 point sources.

Table 4-6

Input Data Formats for $ARSRCE Package

Parameter	Card No.	Card Column	Format	Description
b$ARSRCE##	1	1-10	A2, A6, I2	Identifies input of area source data on next card. The #'s indicate the number of the area source, i.e., $ARSRCE01, $ARSRCE02, ... etc. These numbers must be encountered sequentially in the input stream. b = leading space.
NAMES	2	3-10		Area source name, i.e., quarry 01
Q1	2	11-20	F10.0	Emission rate (g/s) of pollutant 1
Q2	2	21-30	F10.0	Emission rate (g/s) of pollutant 2
XCOR	2	41-50	F10.0	X coordinate of the southwest corner of the area source (km)
YCOR	2	51-60	F10.0	Y coordinate of the southwest corner of the area source (km)
SIDE	2	61-70	F10.0	Side length (km) of the area source square

Note: Area source package (2 cards each) can be repeated for up to 50 area sources.

Table 4-7

Input Data Formats for $CITY Input Package

Parameter	Card No.	Card Column	Format	Description
b$CITYbb##	1	1-10	A2, A6, I2	Identifies input of city source data on next two cards. After the Y in CITY, there are two spaces, then the #'s indicate the number of the city source, i.e., $CITY--01, $CITY--02, ... etc. These numbers must be encountered sequentially in the input stream. b = leading space.
NAMES	2	3-10	2A4	City source name, i.e., Raleigh
Q1	2	11-20	F10.0	Emission rate (g/s) of pollutant 1
Q2	2	21-30	F10.0	Emission rate (g/s) of pollutant 2
XCOR	2	41-50	F10.0	X-coordinate of the southwest corner of the city source (km)
YCOR	2	51-60	F10.0	Y-coordinate
SIDE	2	61-70	F10.0	Side length (km) of the city source
SZ0	3	11-70	6F10.0	Initial σ_z (m) for stability classes A through F, respectively

Note: City source package (3 cards each) can be repeated for up to 5 city sources.

E. $METCTL

The $METCTL package indicates which days of the year are to be processed and if so, from which files to get the meteorological data. It also provides the number of precipitation files that are to be used and their logical unit numbers, sequentially numbered from 61 to 65. It also provides the user with an option to print the meteorological data. The threshold temperature for frozen/ liquid precipitation is also specified in this package.

The $METCTL package is explained in detail in Table 4-8.

F. $CATCHS

The $CATCHS package identifies and associates each receptor point in the study area grid to a catchment and associates the rainfall data sets to the catchments. Free format is used. In free format, inputs are separated by commas and there must be an input for each variable, even when the value is 0.

The $CATCHS package is explained in detail in Table 4-9.

G. $ENDATA

$ENDATA tells DiDOT that all required input data has been read in, and not to expect any more. It is placed after the end of the other input data packages. When DiDOT encounters $ENDATA, it stops reading inputs, and begins to process the data.

4.3.1.3 USER INPUT TO THE POSTPROCESSOR

There is one user input file to the DiDOT postprocessor. If the first record (read in I1 format) contains a 0, no further records will be read from the file. If the first record contains a 1, then the postprocessor will read in the matrix PFACIN (2, 12, 5) of monthly background potency factors for two pollutants, twelve months, and five years in F10.6 format.(Since the present system is designed to handle only one pollutant, every second input number is meaningless, but must be included nevertheless. Furthermore, numbers must be given for five years, beginning with the first year in the study, even if there are fewer than five years in the study.)

4.3.2 NPSDEP

The input data formats for NPSDEP are nearly identical to NPS (see Donigian and Crawford, 1976), such that only differences from NPS inputs are described here.

The NAMELIST/DTYP/card should now include the variables:

DOT = logical flat (T or F) to indicate whether a DiDOT input stream is available (True if DiDOT has been run)

EXAMS = logical flag (T or F) to indicate whether output will be directed to EXAMS (True for subsequent EXAMS run)

ISHED = the number

Table 4-8

Input Data Formats for $METCTL Input Package

Parameter	Card No.	Card Column	Format	Description
b$METCTL	1	1-8	A2, A6	Identifier for meteorological control data package
IDAYS(1)- IDAYS(31)	2	11-72	10X, 31I2*	Options for each day in January. The code follows and is applicable for the entire year -1 = hourly met data will be read from card reader (unit 5) 0 = don't run this day 1 = hourly met data will be read in preprocessed format on logical unit No. 8 Any other digit is not applicable and may be used on Feb. 29 to signify a non-leap year
IDAYS(32)- IDAYS(60)	3	11-68	10X, 29I2	Options for each day in February, including Feb. 29 for all years
IDAYS(61)- IDAYS(91)	4	11-72	10X, 31I2	Options for each day in March
IDAYS(92)- IDAYS(121)	5	11-70	10X, 30I2	Options for each day in April
IDAYS(122)- IDAYS(152)	6	11-72	10X, 31I2	Options for each day in May
IDAYS(153)- IDAYS(182)	7	11-70	10X, 30I2	Options for each day in June

(continued)

Table 4-8 (Continued)

Parameter	Card No.	Card Column	Format	Description
IDAYS(183)- IDAYS(213)	8	11-72	10X, 31I2	Options for each day in July
IDAYS(214)- IDAYS(244)	9	11-72	10X, 31I2	Options for each day in August
IDAYS(215)- IDAYS(274)	10	11-70	10X, 30I2	Options for each day in September
IDAYS(275)- IDAYS(305)	11	11-72	10X, 31I2	Options for each day in October
IDAYS(306)- IDAYS(335)	12	11-70	10X, 31I2	Options for each day in November
IDAYS(336)- IDAYS(366)	13	11-72	10X, 31I2	Options for each day in December
METPRT	14	11-15	I5	Print option for MET. DATA 0 = always off 1 = always on
NPPT	14	16-20	I5	Number of precipitation files (up to 5)
LUNPPT(1)	14	21-25	I5	Logical unit number that precipitation files are found on, beginning with 61, sequentially numbered through 65.
LUNPPT(2)	14	26-30	I5	
LUNPPT(3)	14	31-35	I5	
LUNPPT(4)	14	36-40	I5	
LUNPPT(5)	14	41-45	I5	
TSNOW	15	11-20	F10.0	Threshold temperature for frozen/liquid precipitation (°F).

* The leading 10 spaces may be used to code the name of the month for that card, or abbreviated names of the variables coded on that card.

Table 4-9

Input Formats for $CATCHS Control Package

Parameter	Card No.	Card Column	Format	Description
b$CATCHS	1	1-8	A2, A6	Identifier and geographic data input package.
NNORTH	2	11-15	I5	Number of unit squares in the N-S dimension
NEAST	2	16-20	I5	Number of unit squares in the E-W dimension
INDEX (1) through INDEX (MAX)	3 M		Free Format	Catchment number that each receptor is associated with, up to 3. The subscript of the INDEX array indicates the receptor number. The value of MAX is determined in the program as the product of NNORTH and NEAST (MAX=NNORTH*NEAST). If a receptor square lies in more than one catchment, i.e., at the boundary between two catchments, then the catchment numbers are merged. If catchments are merged, 2 spaces are needed to identify the catchment. For example, if a receptor were in 3 catchments, the input would need 6 spaces, ------, followed by a comma. There would be no spaces between the catchments associated with that one receptor. This continues until all receptors have been associated with the appropriate catchments, using as many cards as necessary. There may be up to 20 catchments, arbitrarily numbered.

(continued)

Table 4-9 (Continued)

Parameter	Card No.	Card Column	Format	Description
KNTEX	M+1	11-15	I5	Number of catchments which are to be used by EXAMS (max = 10)
NEXAMS(1) through NEXAMS(KNTEX)	M+2	1-80	Free Format	Specifies the specific identification number of the catchments to be used by EXAMS
INDPPT(1) through INDPPT(20)	M+3 M+4	11-60 11-60	10I5 10I5	This assigns each of the 20 catchments to a rainfall file on a one-to-one assignment basis. Thus, catchment N will be associated with rainfall file INDPPT(N). There must always be a precipitation file associated with each and every catchment previously referenced in INDEX.

Sample User Input Stream for DiDOT is shown in Figure 4-5.

```
INPUT DATA FOR AIR/WATER INTERFACE MODEL DIDUT
STUDY AREA IS THIRD FORK CREEK, DURHAM, NORTH CAROLINA
 $CNTROL
US,X,Y,RUF  0.5          687.6      3982.3    1.5
TAU        99999.99       10.0
VSETTL      0.004        0.0004
VDEPOS      0.02         0.0075
SCAVNG      5.0E+05      6.0E+04
BKGRND     10.0          1.0E-04
 $METCTL
JANUARY        -1-1-1-1     -1-1
FEBRUARY
MARCH
APRIL
MAY
JUNE
JULY
AUGUST
SEPTEMBER
OCTOBER
NOVEMBER  -1
DECEMBER
PR,N,LUNS      1     1    61
TSNOW       32.0
 $CATCHS
NUP,NACR       5     6
19,1,1,1,19,
1,1,1,2,2,
1907,1,1,1,2,
19,1,1,2,2,
19,1,2,2,2,
19,19,1,2,19
EXAMS KNT.     1
7
INDP: 1-10    61    61    61    61    61    61    61
INDP:11-20                                                 61
 $PTSRCE01
  L AND M 1.07          1.46E-04          687.6     3984.3
          50.           2.0               461.      10.4
 $PTSRCE02
  DUR HOS .147          3.64E-05          689.2     3990.0
          11.       .8           5.2      400.
 $PTSRCE03
  AMERTOBC1.73          5.10E-04          688.8     3985.0
          61.       2.4                   446.      12.5
 $ARSRCE01
  C3M     5.176         5.19E-04          685.6     3986.3    2.
 $ARSRCE02
  E3R     3.64          3.05E-04          685.6     3984.3    2.
 $ARSRCE03
  G4R     2.6           1.61E-04          686.6     3982.3    2.
 $ARSRCE04
  E5C     28.57         3.20E-03          687.6     3984.3    4.
 $ARSRCE05
  C1R     1.68          3.35E-05          683.6     3986.3    2.
 $ARSRCE06
  G6M     5.49          5.63E-04          688.6     3982.3    2.
 $ARSRCE07
  E1R     1.68          3.35E-05          683.6     3984.3    2.
 $ARSRCE08
  H1R     3.78          7.54E-05          683.6     3981.3    3.
 $ARSRCE09
  K1R     3.78          7.54E-05          683.6     3978.3    3.
 $ARSRCE10
  G8R     3.06          2.92E-04          690.6     3982.3    2.
 $ARSRCE11
  B9M     .65           4.03E-05          691.6     3987.3    1.
 $ARSRCE12
  C9M     1.11          1.04E-04          691.6     3986.3    1.
 $ARSRCE13
  D9M     1.39          1.43E-04          691.6     3985.3    1.
 $ARSRCE14
  E9M     1.34          1.36E-04          691.6     3984.3    1.
 $ARSRCE15
  K4R     6.72          1.34E-04          686.6     3978.3    4.
 $ARSRCE16
  K8R     6.72          1.34E-04          690.6     3978.3    4.
 $ARSRCE17
  G10R    1.68          3.35E-05          692.6     3982.3    2.
 $ARSRCE18
  E10R    1.68          3.35E-05          692.6     3984.3    2.
 $ARSRCE19
  C10R    1.68          3.35E-05          692.6     3986.3    2.
$ENDATA
```

FIGURE 4-5

The NAMELIST/YPTM/card is expected to contain an additional variable. PMDVEC (NLAND) the annual potency factor for deposited sediment as a function of land use. The NAMELIST/MPTM/card is expected to contain an additional matrix PMDMAT (12, NLAND), the monthly potency factor for deposited sediment for each land use. The NAMELIST/INAC/card is expected to contain one additional variable SGENI, the initial reservoir of generated fines on pervious areas, while SRERI, given on the same card, takes a new meaning in NPSDEP, i.e., the initial reservoir of deposited sediment on pervious areas.

4.3.3 EXAMS

In batch mode, a number of user inputs are entered in a file which is assigned to logical unit number 5 (SYSIN). The format is identical to that specified for batch operation of EXAMS by Burns, et al (1980) with the following modifications. An additional line is inserted at the head of the file (line 1) containing:

	(Columns)
NPRINT - number of iterations per printout	1-2
NPS - logical flag, T if input coming from NPS	3
DOT - logical flag, T if input coming from DiDOT	4

Following this line are the code names for the chemical and ecosystems (previously lines 1 and 2, now lines 2 and 3). The fourth line of the file contains the variable PERIOD - the load averaging time (hours). PERIOD must be an integer multiple of 24. NPS loadings (output of NPSEXAMS) are daily totals. If PERIOD = 72, 3 (PERIOD/24) daily totals will be averaged by EXAMS, prior to further analysis. Finally, SYSIN contains the system loadings, STRLDG (stream), NPSLDG (NPS), PCPLDG (wet deposition), DRFLDG (dry deposition) and IFLLDG (interflow or groundwater loading). In ALWAS applications, when NPS and DOT are TRUE, NPSLDG, PCPLDG and DRFLDG should be set to zero, unless the user has evidence that toxicant is entering the waterbody by these pathways which is not accounted for by NPS and DiDOT, i.e., unaccounted sources. Loadings specified in SYSIN will be added to those calculated by DiDOT and NPS.

4.3.4 NPSEXAMS

User inputs to NPSEXAMS are limited to:

NDAY - number of days in the simulation period

NSHED - number of NPS watersheds (separate NPS simulations)

MAP $(I)^{1,10}$ - the EXAMS water body which receives runoff from watershed/catchment I

SCALE - the ratio of area between the catchment (delivering water to the EXAMS waterbody) and the NPS calibrated watershed.

These variables are arranged in two card images, thus:

Card I

Variable	NDAY	NSHED	MAP(I)	MAP(2)	. . . . . .	MAP (10)
FORMAT	I4	I2	I2	I2		I2
Columns	1-4	5-6	7-8	9-10		25-26

Card 2

Variable	SCALE(1)		SCALE(2)		. . . .	SCALE(10)	
FORMAT	F6.0	1X	F6.2	1X			1X
Columns	1-6	7	8-13	14		64-69	70

4.4 GUIDANCE IN ESTIMATING INPUT PARAMETERS

ALWAS demands numerous input parameters which may not be available to the user from literature survey, existing data bases, or special observations. Most of these input parameters are needed by NPSDEP. Guidance in parameter estimation and calibration for NPSDEP is provided by the user's manual for NPS (Donigian and Crawford). The only differences in input requirements between NPS and NPSDEP are the deposition rates (referred to as accumulation rates in NPS documentation). The new inputs required by NPSDEP are provided fully by the input of the DiDOT postprocessor, and no additional user inputs are required beyond those discussed by Donigian and Crawford (1976).

Similarly, a discussion of EXAMS input parameters and means to estimate them is beyond the scope of this report. A discussion of EXAMS input parameters is available in Burns, et al. (1980); Smith, et al., 1977, Smith, et al., 1978; and Lassiter, et al, 1978. However, the dynamic version of EXAMS aquatic ecosystem model, which has been developed specifically for use in ALWAS, requires one additional input parameter not required by EXAMS as described by Burns, et al., (1980). That parameter is the load averaging time, which is also the basic period over which EXAMS response is simulated in ALWAS. A rationale for selecting an appropriate load averaging time for the dynamic version of EXAMS is presented in 4.4.5.

Since most NPSDEP and EXAMS input parameters are described in the aforementioned reports, the emphasis of this section is to provide guidance in estimating input parameters required by DiDOT, particularly those chemical specific parameters whose values may not be available in the scientific literature for all pollutants of interest. These include the deposition velocity (discussed in 4.4.1), settling velocity (see 4.4.2) and the precipitation scavenging ratio (see 4.4.3). In addition, there are several parameters which affect deposition through their effect on atmospheric dynamics. Two of these, the atmospheric stability class and the mixing height are determined from meteorological data automatically by the DiDOT preprocessor, CRSMET. However, a specific meteorological data set may be input by hand, thus bypassing the preprocessor. To assist the user in exercising this option, 4.4.5 addresses methods for evaluating these parameters.

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD: BEGIN SECTION HERE

4.4.1 Deposition Velocity

The deposition velocity input by the user is to be considered a nominal value, since DiDOT calculates an hourly deposition velocity as a function of meteorological conditions, depending on the nominal user input. The rationale for DiDOT's hourly deposition velocity algorithm is presented in Section 3.1. An important feature of this algorithm regards the inherent assumption that the deposition of pollutants having observed deposition velocities less than 0.1 cm/sec is limited by exchange through the laminar boundary layer immediately adjacent to the air/ surface interface. For such pollutants, transfer through the turbulent atmospheric boundary layer is always more rapid than that rate through the laminar interfacial boundary layer. When the deposition velocity is less than 0.1 cm/sec, airsurface exchange is limited by the exchange rate of the laminar boundary layer, and will not fluctuate with meteorological conditions.

On the other hand, if a pollutant is observed to have a deposition velocity exceeding 0.1 cm/sec, it is assumed that the exchange rate is limited by the rate of transfer due to atmospheric turbulence. The exchange rate due to atmospheric turbulence fluctuates under different meteorological conditions. Then, if the input nominal deposition velocity exceeds 0.1 cm/sec, the nominal value is disregarded internally by DiDOT which computes the true deposition velocity consistent with the hourly meteorological conditions. This feature of the algorithm implies that DiDOT is insensitive to the precise value of the input nominal deposition velocity so long as the nominal value exceeds 0.1 cm/sec. Results will be the same if the input is 0.5 cm/sec or 5 cm/sec. This is an important consideration pertaining to the estimation of the nominal input deposition velocity.

Vapor/Aerosol Partitioning

Deposition velocities for most toxic organic vapors are expected to be extremely small because of their high molecular weights, and resultant low rates of diffusion through laminar interfacial boundary layers. This phenomenon will be discussed in detail under the heading "Vapor Deposition Velocity". Consequently, the predominant pathway for air-to-surface transfer of toxic organic gases will be via adsorption to ambient aerosols with subsequent deposition of the aerosol. (Eisenreich, et al., 1981). To quantify this phenomenon, one must first estimate the partitioning of airborne contaminant between vapor and sorbed forms. Junge (1977) presented a formula for evaluating this partitioning, given by:

$$\phi = \frac{.165\,\Theta}{p_o + .165\,\Theta} \qquad 4\text{-}1$$

Where Θ is the adsorbed fraction of the total airborne mass, Θ is the available aerosol surface, and Po is the saturation vapor pressure of the contaminant at ambient temperature. This functional relationship is plotted in Figure 4-6, along with representative ranges of Θ .

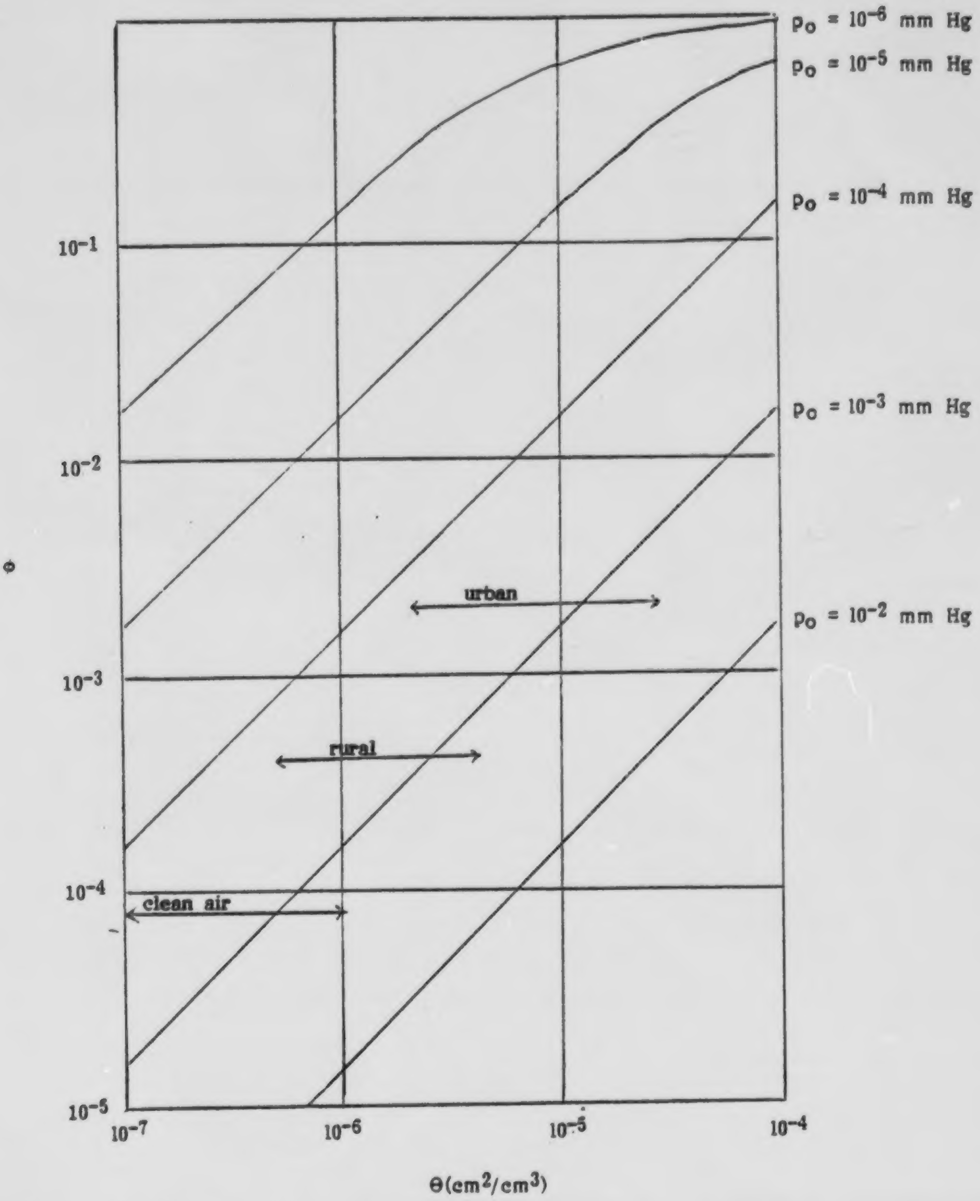
BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA, OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"

PAGE NUMBER
EPA-287 (Cin.)
(4-78)

FIGURE 4-6

DEPENDENCE OF VAPOR/AEROSOL PARTITIONING ON CONTAMINANG VAPOR PRESSURE

$\phi(p_o, \Theta)$ for $\phi < 10^{-1}$



The fraction of the airborne contaminant reservoir which is sorbed to aerosols, ϕ, may be estimated as a function of contaminant vapor pressure, ρ_o, at ambient temperature and the aerosol surface area concentration, Θ, using this graph based on the formula presented by Junge (1977). For example, if $\rho_o = 10^{-4}$ mm Hg, and $\Theta = 10^{-5}$ cm^2/cm^3, a representative value for urban areas, then $\phi \approx 1.5 \times 10^{-2}$, i.e., 1.5% is sorbed, 98.5% vapor. When $\rho_o << 10^{-6}$ mm Hg, $\phi > 10^{-1}$ for all but the cleanest air masses, and deposition processes will be controlled by the sorbed phase. For details of the dependence of ϕ on ρ_o and Θ, $\rho_o << 10^{-6}$ mm Hg, consult Junge (1977).

Given a partitioning among gaseous and adsorbed fractions, the deposition velocity for the total mass of contaminant is given by:

$$W_d = W_{d,v}(1-\phi) + (W_{d,s} (\phi) \qquad 4\text{-}2$$

where $W_{d,v}$ is the deposition velocity for the vapor fraction, and $W_{d,s}$ is the deposition velocity of the sorbed fraction.

Estimation of $W_{d,v}$

The dry deposition velocity for the gaseous fraction may be estimated by the following:

$$\log W_{d,v} = -.383\ [(.0225/D_a) + H/(23.8\ D_w + .042\ K_w)] + .576 \qquad 4\text{-}3$$

where

$D_a D_w$ are the molecular diffusion coefficient in air and water, respectively (cm^2/sec);

H is the Henry's Law Constant (atm/mole fraction); and

K_w is the total pseudo-first order reaction rate constant for the contaminant in water (sec^{-1}).

This formula is based on a semi-empirical analysis. In order to develop a simple method for estimation of the deposition velocity, very broad simplifying assumptions were taken. Foremost is the assumption that surface deposition processes are controlled by the interaction of the contaminant with surface water. This water mass may be found in plant tissues or as soil moisture, or as a body of water. This hypothesis has been suggested by earlier investigators (Hill & Chamberlain, 1974), although there surely exist some pollutants for which it is not true. Use of this assumption as a working hypothesis is further bolstered by the collection of deposition data presented by Peterson in 1977 which shows that deposition velocities over land and water are very nearly equal for many contaminants. Further, it is assumed, merely for the simplicity of a general approach regardless of surface, that the fluid dynamics of the surface are the same as an open body of water stirred by the wind. This is analogous to the study of actual evaporation rates by estimating potential evapotranspiration. Finally, since DiDOT is especially suited to evaluate the effects of fluctuating meteorological conditions on the deposition velocity, we are interested here primarily in the variation of W_d associated with variation of chemical properties as opposed to weather conditions. Consequently, it is assumed that the transfer velocity through the lower atmosphere is given by the climatological average of 1 cm/sec.

On this basis, theoretical expressions for the deposition velocity were derived, based primarily on the multi-layer approach presented by Slinn, et al. (1978) which yields:

TYPING GUIDE SHEET
EPA-287 (Cin.)
(4-76)

$$\frac{1}{W_{d,v}} = \frac{1}{k_g} + \frac{H}{k_l}$$

k_g is the exchange coefficient for air side,

k_l for the liquid side of an interface

The multi-layer theory is based on a model of the air-water interface represented by Figure 3-3. For a contaminant to pass from air to water, it must be transferred by atmospheric turbulence from a convenient reference height (where the concentration C_{air} is measured) to the laminar air boundary layer, by molecular diffusion through that boundary layer to the air water interface where air and water concentrations are assumed to be equal, given Henry's law equilibrium. Molecular diffusion through the liquid phase laminar boundary layer carries the contaminant to the turbulent bulk of the liquid phase. In traveling through these four layers, the exchange rate will be controlled by the layer having the lowest exchange rate, which is analogous to a bottleneck.

$$\frac{1}{k_g} = \frac{1}{k_{at}} + \frac{1}{k_{gvsl}}$$

where k_{at} is the exchange rate for transfer by atmospheric turbulence (1cm/sec) and k_{gvsl} is the rate for transfer through the gaseous viscous sub-layer adjacent to the air water interface.

$$\frac{1}{k_l} = \frac{1}{k_{lvsl}} + \frac{1}{k_{wt}}$$

where k_{lvsl} is the exchange rate for the liquid viscous sub-layer and k_{wt} is the exchange rate for the bulk water phase (turbulent).

If $k_{at} \sim k_{wt} \gg k_{gvsl} \sim k_{lvsl}$, then the bottleneck is in the interfacial boundary layers. This condition is likely to be obtained for many organic vapors under typical meteorological conditions, so the emphasis is on the interfacial exchange rate, defined as:

$$\frac{1}{k_{int}} = \frac{1}{k_{gvsl}} + \frac{H}{k_{lvsl}}$$

and

$$k_{lvsl} = D_w/\delta_l$$

where D_a and D_w are the molecular diffusion coefficients of the contaminant in air and water, respectively, δ_l is the thickness of the sublayer, and k_w is the pseudo-first order reaction rate for the contaminant in water. Expressions for δ_g and δ_l are given by Krauss (1972) for a wind-stirred air-water interface as:

TYPING GUIDE SHEET
BEGIN FIRST LINE OF TEXT HERE
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

$$\delta_g = 75 \ \vartheta_a/U \qquad \delta_1 = \frac{75\,\vartheta_w}{U} \sqrt{\varrho_w/\varrho_a}$$

where ϑ_a and ϑ_w are the kinematic viscosities of air and water, respectively ϱ_a and ϱ_w, the respective densities and U the wind speed at instrument height (3-10m).

Combining these definitions, taking the log, and inserting values for ϑ_a = .15 cm/sec, ϑ_w = .01 cm^2/sec, U = 500 cm/sec, yields

$$-\log K_{int} = \log\left[\frac{.0225}{D_a} + \frac{H}{23.8D_w + .042k_w}\right]$$

where a typical wind speed was used to estimate the nominal deposition velocity

Available data on the deposition velocity of gases presented by Peterson (1977) was evaluated in order to determine k_{int}. For several gases having observed deposition velocities in the range .1-10 cm/sec, it was concluded that the "bottleneck" was transfer through the bulk turbulent atmosphere and they were assigned $k_{int} \gtrsim 10$ cm/sec.

Application of this formula to the data for 15 gases (Peterson, 1977) did not result in good predictive capability. Nonetheless, the quantity;

$$\log_{10}\left[\frac{.0225}{D_a} + \frac{H}{23.8\ D_w + .042\ k_w}\right]$$

was significantly correlated with $W_{d,v}$ (r = .78), so empirical analysis was used to improve the predictive power of the formula. Regression analysis indicated that the formula given at the beginning of this discussion provided the best fit between these parameters. The accuracy is still quite limited, as shown in Figure 4-7 which compares observed and estimated values of the deposition velocity for 15 gases. The standard deviation of the residuals about log V_d is 0.65, implying that predictions are only good with 90% confidence to within a factor of 12. For cases in which the dry deposition of gaseous pollutant is the primary removal mechanism, this level of uncertainty will also apply to the overall deposition velocity. However, if $W_{d,v}\ (1 - \phi) << W_{d,s}\ (\phi)$, then uncertainty in $W_{d,v}$ will be inconsequential. Note that the equation implies that the reaction rate is insignificant unless $K_w > 10^2 D_w$. This condition is met only for NO_2, SO_2, NH_3, and HF.

Dry Deposition Velocity Estimation for the Sorbed Contaminant

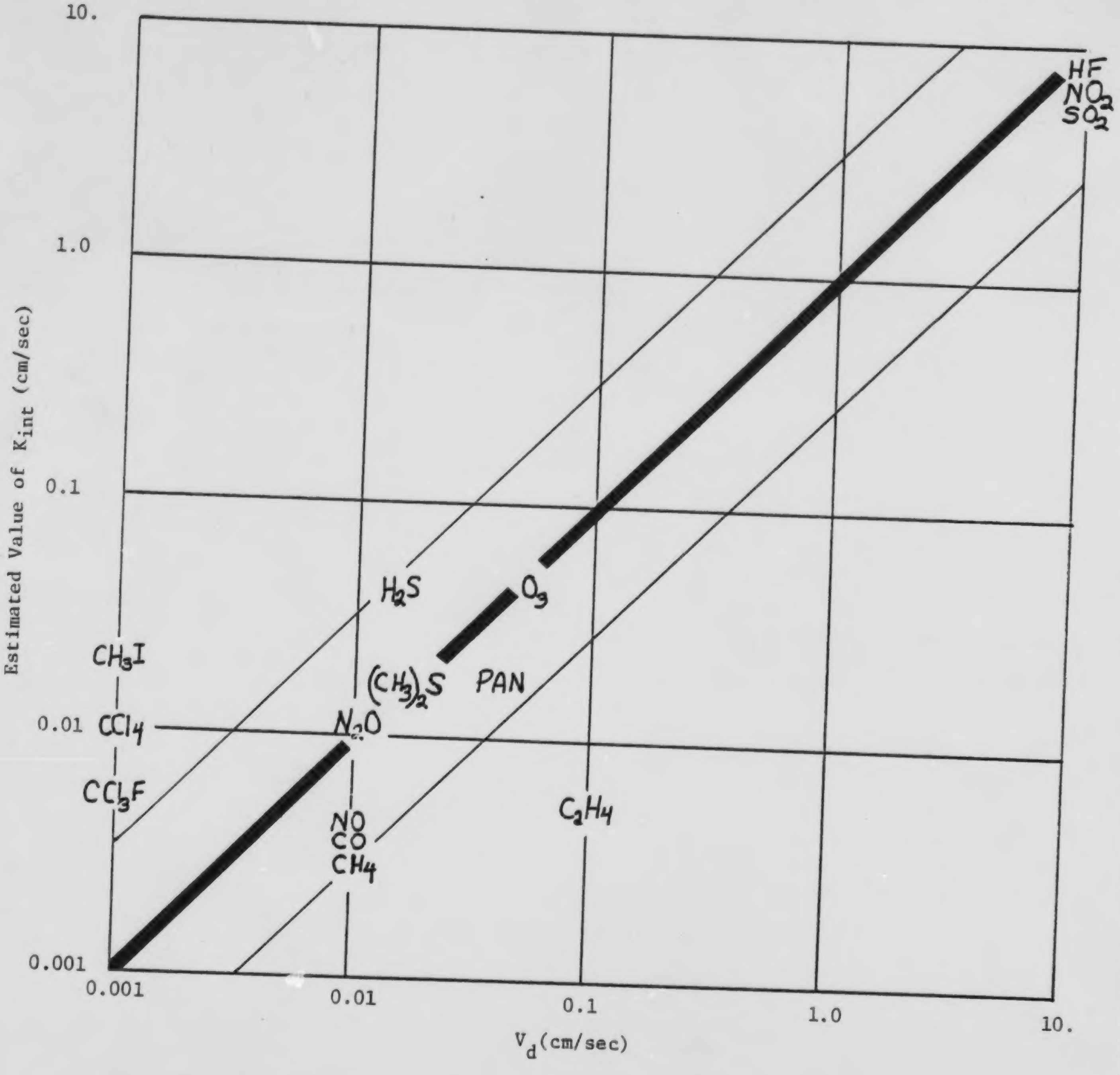
For the fraction of airborne contaminant which is adsorbed, ϕ, the dry deposition velocity $W_{d,s}$, is solely a function of the particle size. The dry deposition velocity of particles has been studied comprehensively by Sehmel and Hodgson

BOTTOM OF IMAGE AREA, OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
BEGIN LAST LINE OF TEXT
3/8"
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

FIGURE 4-7

4-~~8~~ 7

CORRESPONDENCE BETWEEN ESTIMATED AND OBSERVED GASEOUS DEPOSITION VELOCITIES



The heavy diagonal represents perfect prediction. The parallel lines encompass probable error bands for observed V_d. Vertical distance to the nearest outer line is prediction error. The average absolute error is a factor of 3, with 90 % confidence of being within a factor of 12 .

52

91

TYPING GUIDE SHEET

(1974), who developed a series of nomographs for the deposition velocity as a function of particle size under various meteorological conditions. Wind speed, surface roughness, and atmospheric stability conditions were found to be the meteorological parameters which affected the deposition velocity. Our goal is to define the deposition velocity under typical meteorological conditions, for use as a nominal input, avoiding the complications of minor variations (less than an order of magnitude) due to meteorological conditions. Reviewing all of Sehmel and Hodgson's results, it is concluded that the functional dependence of deposition velocity upon particle size across all meteorological condition can be represented within an order of magnitude by Figure 4-8, which indicates deposition velocity $W_{d,s}$, of 1 cm/sec for all particles having diameter less than 20 cm, and with larger particles being deposited at a velocity equivalent to the Stokes settling velocity (see Section 5.4.2).

Effect of Source Type

Five source types are believed to be unique with respect to the physical form of the emitted contaminant. Source Type 1 involves the release of a vapor into the ambient air. The vapor would then equilibrate via condensation and adsorption with the ambient aerosol according to its vapor pressure, as discussed above under "partitioning". The key to evaluating the physical form and deposition velocity from sources of Type 1 is the use of the properties of the ambient aerosol for partitioning with the vapor. In order to perform deposition estimates for sources of this type, one must first determine the partitioning between vapor and sorbed phases. Refer to Figure 4-6 or equation (5) to estimate Φ. If released into an urban area, the ambient aerosol will have a surface area, of 2×10^{-6} to 2×10^{-5} (cm^2/cm^3) as shown in Figure 1. A median value of $\theta = 6 \times 10^{-6}\ cm^2/cm^3$ is suggested. Similarly, Figure 4-6 shows the appropriate value of Φ for rural and pristine locations. As shown in Appendix B, the adsorbed contaminant will be associated with sub-micron aerosols, ($.1 < d_a^{50} < .2um$) regardless of other environmental conditions, so $W_{d,s}$ is shown in Figure 3-6 to be approximately 1 cm/sec. The dry deposition velocity for the vapor phase is calculated by equation 4-3. Application of equation 4-2 yields W_d.

Source Type 2 is characterized by contaminants released from a combustion zone. The vapor phase is assumed to equilibrate with an aerosol having surface area and size distribution typical of the fly ash contained in stack gases of a coal-fired power plant. It is recognized that not all pollutants arising from combustion processes come from coal-fired power plants, but coal fly ash has been comprehensively studied, and certain fundamental information is available which meets the need of this estimation method. If the user has information regarding specific source characteristics which permit a more specific analysis, the use of such auxiliary data is strongly recommended, based on the methods presented above and in Appendix B. For coal fly ash, Appendix B indicates that $\Theta = 0.9\text{-}1.5\times10^{-3}\ cm^2/cm^3$ and $d^{50} = .1$ m for controlled emissions, $\Theta = (.022\text{-}.075)\ cm^2/cm^3$ and $d^{50} = {}^{a}(.5\text{-}1.4)$ m. It is assumed, without strong supporting evidence, that the gaseous pollutant equilibrates with fly ash in the combustion zone, stack and concentrated initial plume, and subsequently travels with the fly ash aerosol with which it was originally associated. This assumption loses its appeal the further the pollutant travels from its source. At large distances, it may have nearly come to an equilibrium with the ambient aerosol. In that extreme, Source Type II is identical to Source Type I. Analyses using both Release Mode II and Release Mode I methods

EPA-287 (Cin.)
(4-76)

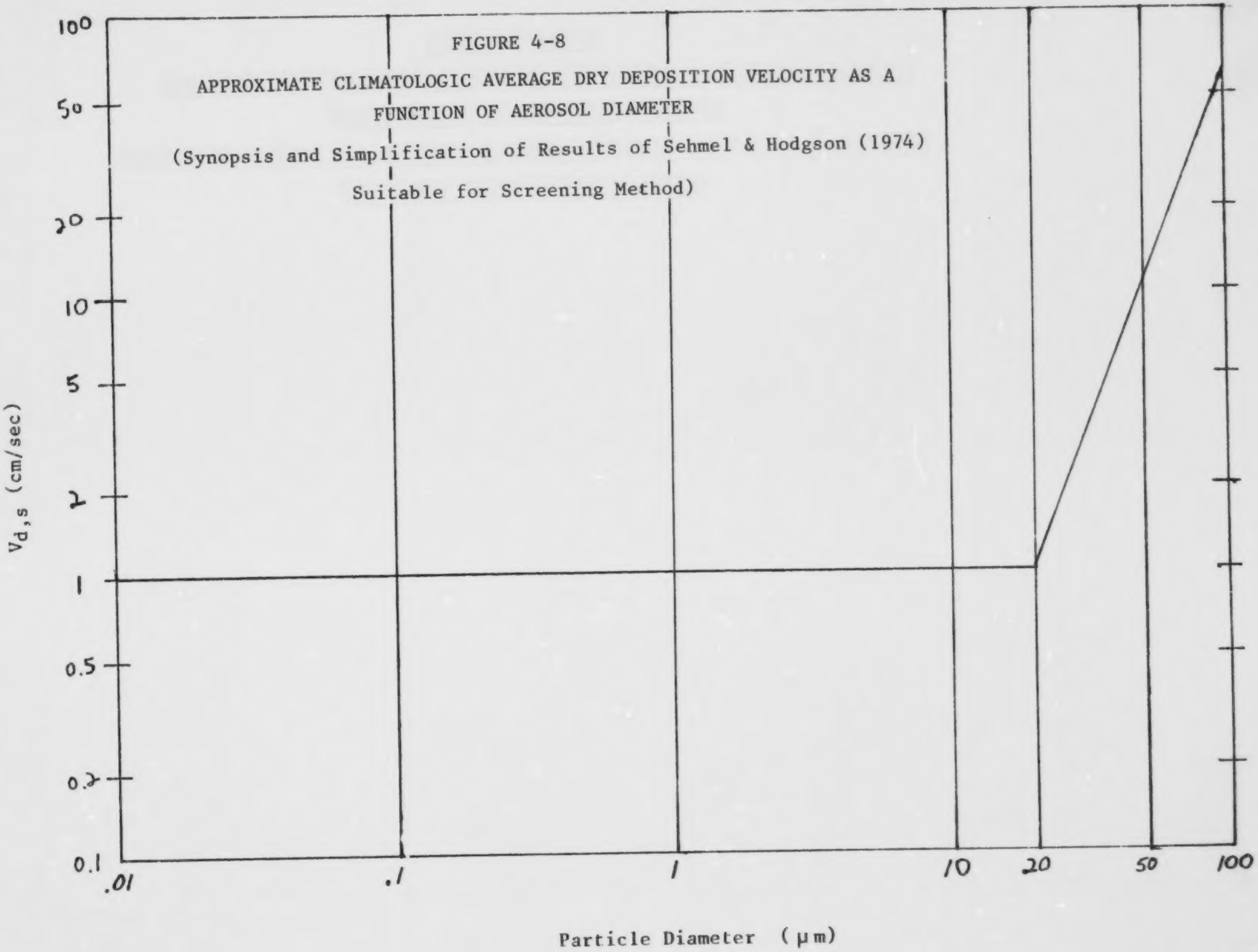


FIGURE 4-8

APPROXIMATE CLIMATOLOGIC AVERAGE DRY DEPOSITION VELOCITY AS A FUNCTION OF AEROSOL DIAMETER

(Synopsis and Simplification of Results of Sehmel & Hodgson (1974) Suitable for Screening Method) are advised for pollutants released from a combustion zone. Differences in the results of the two methods should not be interpreted as upper and lower bounds of the analysis, but rather as the sensitivity to the assumption of reversibility of the adsorption/desorption phenomena. For the sorbed fraction, $W_{d,s} = 1$ cm/sec, since it is assumed the $V_{d,s} = 1$ cm/sec on a long-term average for aerosols with diameter less than 20 m.

To summarize, for Source Type 2:

1. Gather data for the saturation vapor pressure at STP, Henry's Law Constant, molecular diffusion coefficient in air and water, D_a and D_w, and the pseudo first order reaction rate constant in water, k_w (note: if it can be shown that $k_w << 578\ D_w$, the exact value of k_w is unimportant.

2. Use equation 4-1 to determine ϕ. If Θ unknown, use .05 cm^2/cm^3 for uncontrolled fly ash or 1.2×10^{-3} for controlled fly ash.

3. If $d_A^{50} < 20$ m, $W_{d,s} \simeq 1$ cm/sec. For fly ash, this condition is satisfied.

4. Calculate $W_{d,v}$ using Equation 4-3.

5. Calculate W_d using Equation 4-2.

Source Type 3

Source Type 3 applies for contaminants such as pesticides which have been well mixed with soils which are subsequently stirred up by winds, vehicles or earth-moving equipment. It is beyond the scope of this study to review the methods for estimating emissions from such activities. These methods were investigated by Cowherd, et al., (1974). Here we are primarily concerned with estimating the deposition velocity. It has been established that re-entrained dusts exhibit a similar size distribution with the source soil, except that particles larger than 60 μm (sand, gravel, pebbles) are quickly redeposited and are not widely dispersed. Based on Cowherd's studies, it is assumed that silt and clay fractions of the soil are entrained in proportion to their abundance in the native soil. Then, following methods presented in Appendix B, Cowherd's data indicated $d_A^{50} = 0.5$ m and $V_{d,s} = 1$ cm/sec. To estimate the adsorbed fraction of the airborne contaminant use a value of Θ appropriate to the application site, i.e., rural or urban from Figure 4-4.

Source Type 4

When a contaminant is released by abrasion from a coherent solid, as for example, in rock crushing, mining, materials handling, a much larger diameter is appropriate, ranging from 30 - 100 m. The dry deposition velocity is given by Stokes law (see following section).

Stokes settling velocity for a particle density of 1.5 gm/cm^3 is approximately 4 cm/sec at $d = 30\ \mu m$ and 35 cm/sec at $d = 100\ \mu m$.

Source type 5 applies to particulate contaminants of known size distribution. When the particle size of the contaminant is known a priori, the deposition velocity can be estimated using methods described for previous source types.

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

The methods presented in this section have been shown to be valid, and accurate to within an order of magnitude, for a wide variety of pollutants and source types in work performed by Arthur D. Little, Inc., under Task V of contract 68-01-5949 with the EPA - Office of Water Planning and Standards (Arthur D. Little, Inc., 1981).

4.4.2 Settling Velocity

The settling velocity of particulate contaminants is given by Stokes law, i.e.,

$$W_s = 8/9\ gd^2\rho$$

where ρ is the particle density (gm/cm^3);
g is the acceleration of gravity (9.8 cm/sec^2);
d is the diameter (cm); and
V_a is the kinematic viscosity of air.

Particulate deposition velocity should always be greater than, or equal to, the settling velocity, unless particulate resuspension is occuring. The resuspension of particles is a phenomenon which is not adequately described by available theories. ALWAS is not capable of describing resuspension, as a result of deficiencies in both DiDOT and NFSDEP with respect to this phenomenon. This is certainly a fruitful topic for continuing research and development. However, there is a crude method by which ALWAS may be used to evaluate, to first order, the effects of resuspension. This could be accomplished by inputting a deposition velocity which is smaller than the settling velocity. The difference between the true deposition velocity and the input deposition velocity may be characterized as an analogously defined resuspension velocity (i.e., the resuspension flux is the resuspension velocity x the air concentration). However, DiDOT computes its own deposition velocity when the nominal input deposition velocity lies between 0.1 and 10 cm/sec, so this method of accounting for resuspension would not be effective if the true deposition velocity and the modified deposition velocity accounting for resuspension both lie within the aforementioned range.

4.4.3 Scavenging Ratio

In order to estimate the scavenging ratio, first determine the source type and vapor/aerosol partitioning as discussed in 4.4.1. Then the scavenging ratio is given as

$$r = r_s\ \phi + r_v\ (1-\phi)$$

where r is the overall scavenging ratio for the airborne contaminant,
r_s and r_v are the scavenging ratios of the adsorbed and vapor phase, and
ϕ is the adsorbed fraction of the total airborne mass

The vapor phase scavenging ratio is simply the inverse of the Henry's law constant, i.e.,

$$r_v = 1/H$$

where the appropriate units for the Henry's law constant is the dimensionless weight/volume (air) divided by weight/volume (water), e.g., $\frac{ng/m^3}{\mu g/l}$ The scavenging

BEGIN LAST LINE OF TEXT
3/8"
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
PAGE NUMBER
EPA-287 (Cin.) (4-76)

3694

ratio for the aerosol phase has been shown by Gatz (1976) to be a function of particle size. Gatz (1976) data have been plotted in dimensionless form in Figure 4-9. The appropriate size of the adsorbed contaminant is a function of source type (see 4.4.1).

4.4.4 Surface Roughness Length

The surface roughness length may be estimated for various types of terrain and surface cover with reference to Figure 3-1.

4.4.5 EXAMS Load Averaging Time

The interval over which NPSDEP and DiDOT loadings to an EXAMS water body are averaged is determined by EXAMS response time to a change in load. This is most clearly indicated by running the classic version of EXAMS which determines the steady state half-life and related system self-purification time. In order to ensure that all significant temporal fluctuations in water quality are represented in the model response, the load averaging time should be significantly less than the half-life. If the load averaging time was 7.5% of the half-life, fluctuations of 5% or more would be faithfully simulated, while if the averaging time were 15% of the half-life, concentration fluctuations of more than 10% would be faithfully simulated. Thus, a load averaging time of roughly 10% of the total half-life is appropriate.

4.4.6 Fugitive Dust Emission Rates

Fugitive emissions generally encompass the following types of area sources:

- wind erosion from cropland or other unvegetated and exposed surfaces;
- agricultural tilling;
- paved and unpaved roads;
- construction activity;
- aggregate handling and storage.

The emission factors for these sources can be estimated from empirically derived equations.[1,2,3] The empirical equations are based on measurements that were made for equivalent types of sources. These measurements included the use of hi-vol samplers, some equipped with cascade impactors to differentiate particle sizes. The results of these measurements indicated that the mean particle size would be approximately 30 μm for most of the open dust sources and that significant deposition would occur near the sources. When modeling the transport and deposition of the open source fugitive emissions several ranges of deposition velocities should be considered to reflect the various size ranges. Table 4-10 shows the weight fractions of particles in three size classes that would be emitted from the abovementioned open area sources.

---

[1] Woodruff, N.P., et al. (1972)

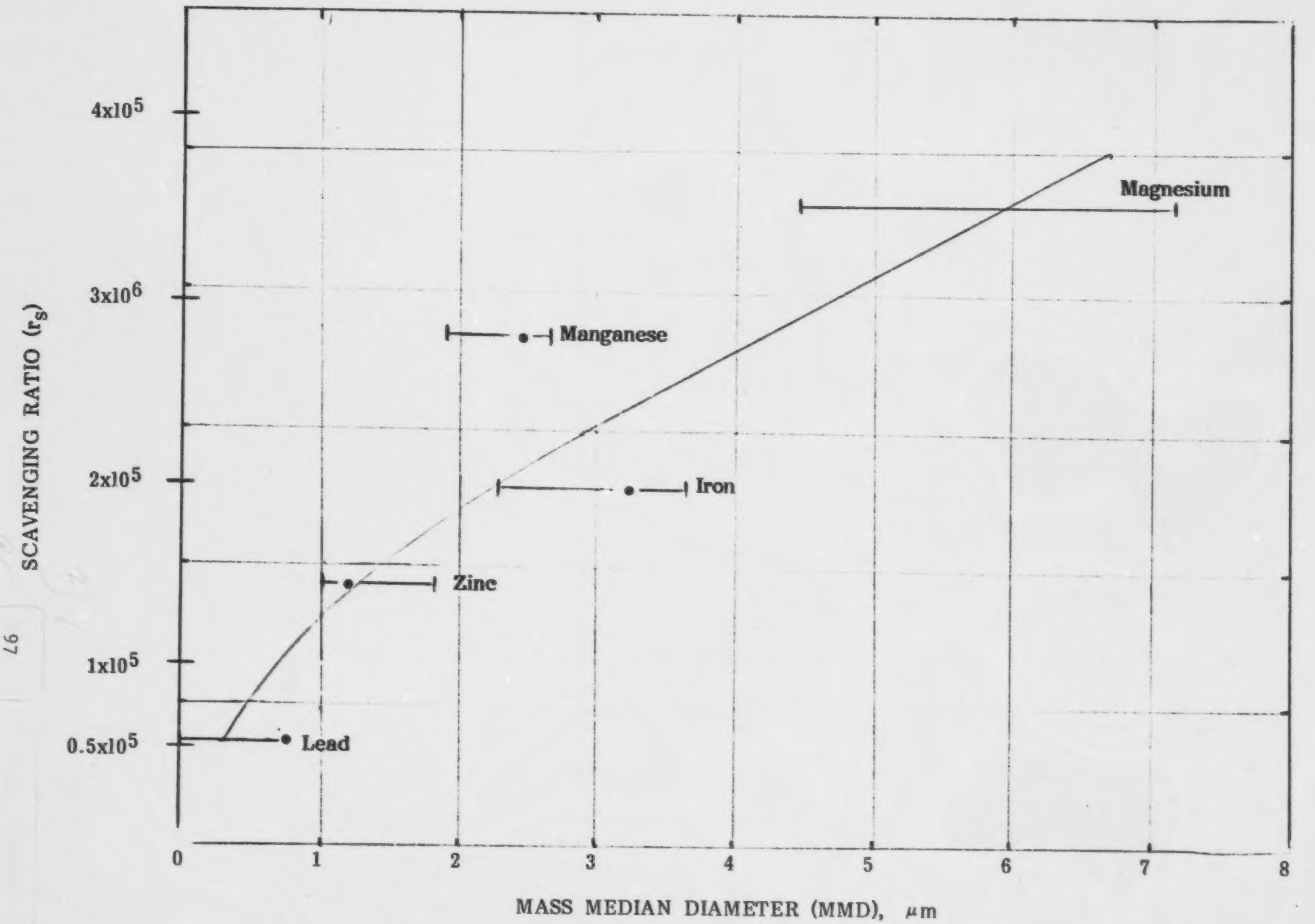
[2] Cowherd, C., et al. (1974)

[3] NTIS Publication (1977)

95

FIGURE 4-9 ~~4-8~~ 4-9

Variation of Scavenging Ratio with Particle size



(adapted from Gatz, 1976).

TYPING GUIDE SHEET
BEGIN FIRST LINE OF TEXT HERE
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD BEGIN SECTIONS HERE

TABLE 4-10

PARTICLE SIZE RANGES FOR OPEN DUST SOURCES*

Source	wt % 2 μm	wt % 2-30 μm	wt % 30-100 μm
Cropland erosion and agricultural tilling (sandy loam)	30	20	50
Unpaved roads	26	31	43
Aggregate handling and storage	32	16	52

*No data exist for construction activity.
The type of construction activity determines the particle size range; i.e., land clearing and excavation would have a soil type particle range, and building destruction might have a particle size range similar to stone crushing for aggregate handling and storage.

BEGIN LAST LINE OF TEXT
3/8"
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

The particle size ranges of these sources are significantly different than emissions from stacks, particularly from combustion sources stacks, and should be treated differently in an atmospheric transport model.

A capability for modeling different size ranges of the total suspended particulate in DiDOT has not yet been developed. The general structure of the model does not preclude further development in this area, but some software modification would be required. In the current version, the total suspended particulates are simulated as if they were all the same size. In order to achieve the best simulation of total suspended particulate transport and deposition, it is recommended that the user adjust emission rates of fugitive dust sources such that the emission rate reflects only the smaller particles, i.e., those particles smaller than some diameter, d_{up}, which have a mass median diameter similar to the total suspended particulate and/or major sources of the modeled region. The typical mass median diameter of American urban area ranges from 2-10 μm. For the fugitive dust sources identified in Table 4-10, then, particles above 30 um should be ignored when preparing the emissions inventory. 30 μm particles released from fugitive dust sources will settle out within about 100 meters from their source under typical meteorological conditions. Since DiDOT will generally be implemented with inter-receptor distances exceeding 100 meters, the exclusion of particles exceeding 30 μm is also recommended since their transport is limited to length scales smaller than the model will resolve.

4.5 USER OPERATING INSTRUCTIONS

This section describes in greater detail those specific aspects of ALWAS components that the user must be aware of in the preparation of his inputs and job control commands to insure proper execution of the system. The discussion below covers DiDOT, its postprocessor, NPSDEP, EXAMS, and the NPSEXAMS interface. It does not include the DiDOT meteorological preprocessor, or the routines which will be necessary to break up and reform long records for inter-computer telecommunications transfer purposes. Where appropriate, examples of input files, output files, and job control streams are given. This section also contains information on the overlay structures of DiDOT and EXAMS.

4.5.1 DiDOT

4.5.1.1 DiDOT OVERLAY STRUCTURE

The overlay structure of DiDOT is shown graphically in Figure 4-10(a) and (b) and its translation into Overlay Description Language is given in Figure 4-11. This overlay structure is optimized for the organization and sequential processing nature of DiDOT. Initialization and input routines are collected into two overlays which are called only once. Obtaining the meteorological data and presetting concentration and deposition arrays for background levels is done once per day, so these are in a single overlay. Cycling over sources occurs hourly within each source category, and so the routines necessary for each source category are loaded together once per day. Since city sources and point sources are treated somewhat similarly and use many of the same routines, those routines which are different are loaded at an overlay level "lower" than those they have in common, as indicated by the COMPS branch of the overlay structure.

EPA-287 (Cin.)
(4-76)

USE THIS SHEET FOR REPORT TYPING 8-1/2 x 11

TOP

10 Lines to End of Typing Area

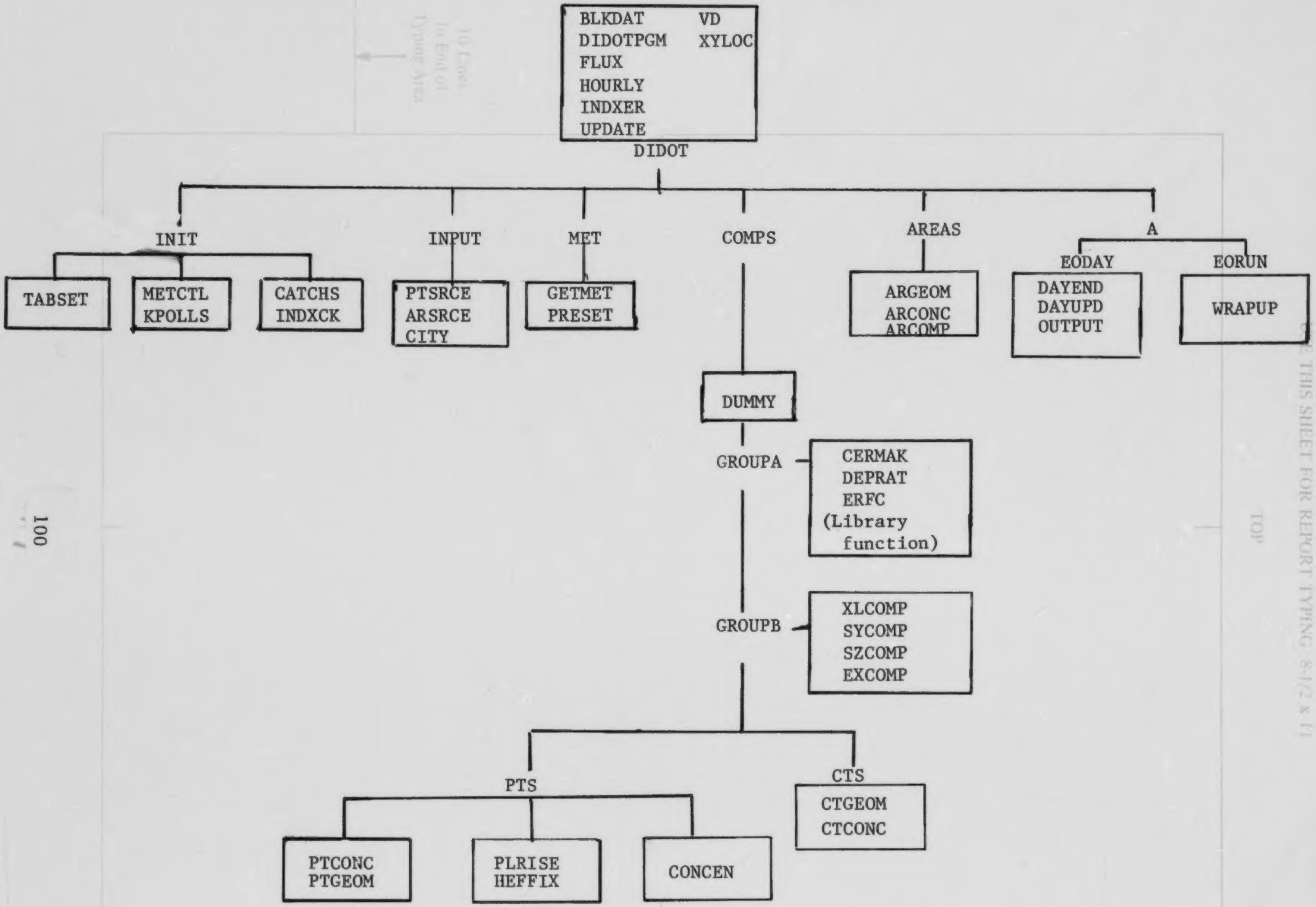

BLKDAT
DIDOTPGM
FLUX
HOURLY
INDXER
UPDATE
VD
XYLOC
DIDOT
INIT
TABSET
METCTL
KPOLLS
CATCHS
INDXCK
INPUT
PTSRCE
ARSRCE
CITY
MET
GETMET
PRESET
COMPS
DUMMY
GROUPA
CERMAK
DEPRAT
ERFC
(Library function)
GROUPB
XLCOMP
SYCOMP
SZCOMP
EXCOMP
PTS
PTCONC
PTGEOM
PLRISE
HEFFIX
CONCEN
CTS
CTGEOM
CTCONC
AREAS
ARGEOM
ARCONC
ARCOMP
A
EODAY
DAYEND
DAYUPD
OUTPUT
EORUN
WRAPUP

FIGURE 4-10 (a): DiDOT OVERLAYS

4-7(a)
-10(a)

For Page Numbering

USE THIS SHEET FOR REPORT TYPING 8-1/2 x 11

TOP

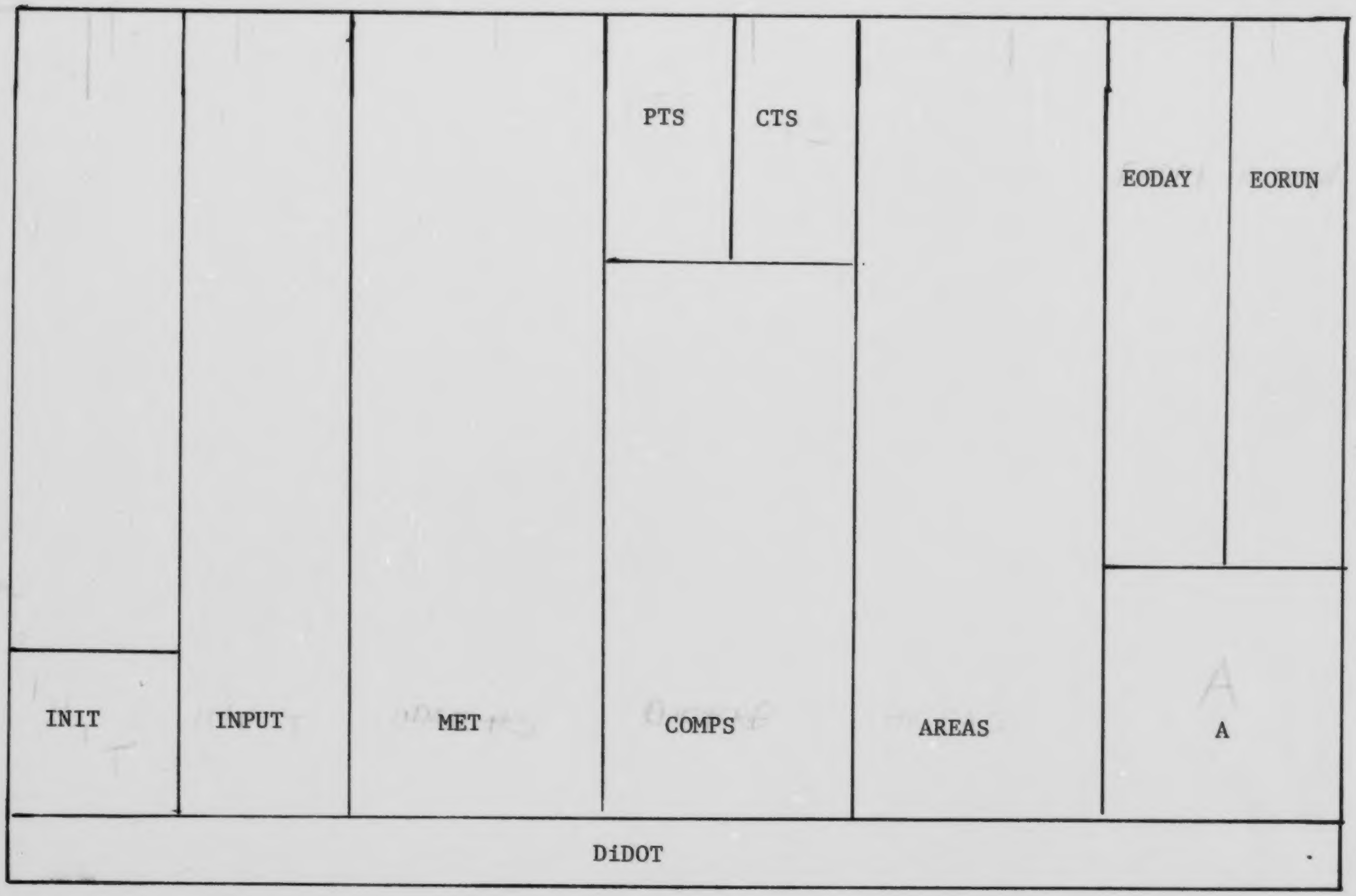

PTS
CTS
EODAY
EORUN
INIT
INPUT
MET
COMPS
AREAS
A
A
DiDOT

FIGURE 4-10 (b): DiDOT MEMORY DIAGRAM

~~4-7~~(b)
4-10

10 Lines
to End of
Typing Area

For Page Numbering

USE THIS SHEET FOR REPORT TYPING 8-1/2 x 11

TOP

10 Lines to End of Typing Area

```
          .ROOT DIDOT-*(INIT,INPUT,MET,COMPS,AREAS,A)
A:        .FCTR EODAY,EORUN
DIDOT:    .FCTR BLKDAT-DIDOTPGM-FLUX-HOURLY-INDXER-UPDATE-VD-XYLOC
INPUT:    .FCTR PTSRCE-ARSRCE-CITY
INIT:     .FCTR TABSET,METCTL-KPOLLS,CATCHS-INDXCK
MET:      .FCTR GETMET-PRESET
COMPS:    .FCTR DUMMY-GROUPA-GROUPB-*(PTS,CTS)
GROUPA:   .FCTR CERMAK-DEPRAT-DB0:[200,275]IMSL/LB:ERFC
GROUPB:   .FCTR XLCOMP-SYCOMP-SZCOMP-EXCOMP
AREAS:    .FCTR ARGEOM-ARCONC-ARCOMP
CTS:      .FCTR CTGEOM-CTCONC
PTS:      .FCTR PTGEOM-PTCONC-*(PLRISE-HEFFIX,CONCEN)
EODAY:    .FCTR DAYEND-DAYUPD-OUTPUT
EORUN:    .FCTR WRAPUP
          .END
```

FIGURE 4-11: DIDOT OVERLAY DESCRIPTION LANGUAGE (ODL)

4-8

For Page Numbering

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

Printed and disk output are produced by routines collected into the A branch. The output necessary at the end of each day is prepared by routines in the EODAY group, and the final summary printed output is given by the EORUN overlay.

It should be noted that the expression used for calculating concentration from point and city sources uses the complementary error function, ERFC. This is not a standard PDP 11/70 FORTRAN Library function, but is being called from the IMSL library. This is shown in Figure 4-11 as the last file reference for the GROUPA FCTR directive.

4.5.1.2 DiDOT INPUT AND OUTPUT FILE CONSIDERATIONS

Figure 4-2 and the discussion in Sections 4.2 and 4.3 indicated the input data requirements and program usage. Some rules were given in that discussion concerning the input deck set up, which are collected and expanded in Table 4-11 for completeness and ease of reference.

DiDOT produces a printed output and an unformatted disk data set. The unformatted data set is designed to be used only as input to the DiDOT postprocessor, which is discussed in Section 4.5.1.4. The printed output consists of an "echoing" of all card-input data packages, and presentations of the receptor grid array and precipitation and catchment indexing relationships in a digitized receptor square map format. The only other standard printed output occurs at the end of the run when tables of highest concentrations are given. Optionally, the input meteorological data for each simulation day can be printed if the variable METPRT in the $METCTL package is set to 1.

Figure 4-12 shows example printed output from DiDOT. These examples are drawn from several runs of DiDOT, and so there is not necessarily consistency between the days turned on and the meteorological data which is shown. Also, some parts of the output which would normally appear on separate pages are shown on a single reproduced page. Nonetheless, these examples show the full range of output formats and information produced by DiDOT.

It may be useful to clarify the format of the data given in the last one or two columns of the last two pages of the sample output. These columns provide a packed representation of the DO-loop index values for the variables names in parentheses in the preceding column and associated with the concentration value given at the left. For example, the fifth highest point source hourly concentration for pollutant 2 at any time and at any receptor was $2.678 \times 10^{-3}\ \mu g/m^3$, and this occurred at receptor number 20, from point source number 3 at hour 18 on day 178. Thus, the two columns are unpacked according to an (I2, I2) and (I2, I3) format. Reference to the earlier printed output would reveal the details of the specific source-receptor geometry associated with this concentration value.

Similarly, the fifth highest daily average concentration for pollutant 1 for any catchment or water-body cell is $548.7\ \mu g/m^3$. This was found for catchment 7 on day 241. The unpacking format for this column is (I2, I3).

BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"

PAGE NUMBER
EPA-287 (Cin.)
(4-76)

102

Table 4-11 4-11

REFERENCE RULES FOR DiDOT INPUT DECK SET-UP

- Unlimited number of title or annotation cards, printed on the first page of DiDOT output, and delimited by a blank card.
- Any of the input packages $CNTROL, $METCTL, $CATCHS, or $PTSRCE may come next, in any order. The recommended order would be $CNTROL, $METCTL, $CATCHS, followed by the various emission source inputs.
- The $ARSRCE, and $CITY input packages can be presented in any order following the $CNTROL and $CATCHS input packages, since they are examined with respect to the receptor grid system given by the $CNTROL variables UNIT, XSWCOR, and YSWCOR, and the $CATCHS variables NNORTH and NEAST. This restriction does not exist for point sources.
- Individual point, area, and city source input packages may be intermingled, if the user desires, but the numbering of each source of the same type must be consecutive. For example, $PTSRCE01, $ARSRCE01, $CITY__01, $ARSRCE02, $CITY__02, $PTSRCE02 is a valid sequence, but $CITY__01, $ARSRCE02, $PTSRCE03,... is not.
- Catchment or water-body cell identifiers can be artibrarily assigned in the range 1-20, but one such identifier should be reserved for the "excess" area in the space between the receptor grid and the actual watershed boundary. By assigning the logical unit number associated with this identifier to a null or dummy data set when the postprocessor is executed, no results will be produced for it and it will be removed from the rest of the simulation.
- When merging catchment identifiers for a single receptor to reflect a boundary between three catchments in that unit square, it is imperative that such a merged value not exceed 32767. Since no single catchment identifier can exceed 20, it is sufficient to insure that at any junction of three catchments, at least one of them has an identifier less than or equal to 3, which would be coded in the high-order digit position. It should also be noted that if catchments with identifiers less than 10 are being merged, they must be coded with a leading zero unless they are coded first. Thus, coded value 1907 and 719 refer to the presence of the same two catchments in a single receptor square.

4-11

Table 4-11 (Continued)

- Two cards (20 values) for the INDPPT array must always be included in the $CATCHS input package, even if catchment identifiers 1 through 10 (specified on the first card) or 11 through 20 (specified on the second card) are not defined. Legitimate values for the INDPPT array are 0 or blank if the catchment associated with the array subscript is not defined, or an integer between 61 through 65 if it is defined. A precipitation data set must be linked to the run with the correct logical unit number (LUN) for each different integer that is specified, and these LUN's must be assigned sequentially, beginning with 61.
- The METCTL subroutine makes an initial attempt to read an unformatted identification record from LUN 8, in all cases. If this LUN is not defined or linked to the run at the control language level, DiDOT presumes hourly meteorological data (not including the precipitation data) will be available for each simulated day on LUN 5, beginning immediately following the $ENDATA card. There must be 24 cards for each day "turned on" by a -1 value for the corresponding IDAYS array element. On the other hand, the user is free to override the meteorology for a given day prepared by the preprocessor with a day's worth of data on cards by using both +1 and -1 values in the IDAYS input (and by linking LUN 8, also, of course). The user should note that when manually coding meteorological input, he must provide wind flow vector values, rather than wind direction values, since these are what is prepared by the preprocessor and expected by DiDOT. The wind flow vector is simply the 180° complement of the wind direction, and it specifies where the wind is blowing to, rather than where it is blowing from.
- Setting TNSOW to a very low temperature, i.e., -50°F, effectively bypasses the pollutant deposition with snowfall (dry deposition) feature.
- Columns 31 through 40 on all emission source input cards is not read except for the SZ0 card for each city source. In this case, these columns provide a value of σ_{zo} for stability class C.
- If the calendar year of the simulation is not a leap year, the input value for the IDAYS array for February 29 must be any integer other than 0 or ±1.

End

FIGURE 4-12: <u>EXAMPLE OUTPUTS FROM D1DOT</u>

```
                        INPUT DATA FOR AIR/WATER INTERFACE MODEL DIDOT
                        STUDY AREA IS THIRD FORK CREEK, DURHAM, NORTH CAROLINA

                                                 P O L L U T A N T
                                                 1          2
       HALF-LIFE (HOUR * * -1)                   1.0000E+06     10.000                     TAU
       SETTLING VELOCITY (M/S)                   4.0000E-03     1.0000E-04                 VSETTL
       DEPOSITION VELOCITY (M/S)                 2.0000E-02     7.5000E-03                 VDEPOS
       SCAVENGING RATIO                          5.0000E+05     60000.                     SCAVNG
       BACKGROUND CONC. (UG/M**3)                25.000         1.0000E-04                 BKGRND

        ROUGHNESS LENGTH (METERS)       1.500                                              RUFLEN

       THE FOLLOWING TABLE INDICATES THOSE DAYS WHICH WILL BE USED IN THIS RUN:
MONTH                  DAYS

JANUARY           0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
FEBRUARY          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MARCH             0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
APRIL             0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MAY               0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
JUNE              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
JULY              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
AUGUST            1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
SEPTEMBER         1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
OCTOBER           0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NOVEMBER          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
DECEMBER          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

       THE CODES IN THE ABOVE TABLE MEAN:
         -1 = THIS DAY WILL BE SIMULATED USING MET. DATA FROM CARDS
          0 = THIS DAY WILL NOT BE SIMULATED
         +1 = THIS DAY WILL BE SIMULATED USING PREPROCESSED MET. DATA ON LOGICAL UNIT 8
   ANY OTHER VALUE ON FEBRUARY 29TH SIGNIFIES A NON-LEAP-YEAR CONDITION, AND THE PROGRAM ADJUSTS THE SEQUENCE
```

FIGURE 4-12 (Continued)

```
THE NUMBER OF INPUT PRECIPITATION FILES IS  1.  UNIT NUMBERS ARE:   61

THE THRESHOLD TEMPERATURE FOR FROZEN/LIQUID PRECIPITATION (DEGREES F) IS: -10.00      TSNOW

N. W. S. STATION IDENTIFIER FOR SURFACE MET. DATA:            13722
CALENDAR YEAR OF RECORD OF SURFACE MET. DATA:                  1972
N. W. S. STATION IDENTIFIER FOR MIXING HEIGHT DATA:           12723
CALENDAR YEAR OF RECORD OF MIXING HEIGHT DATA:                 1972

UNIT SQUARE SIZE=   0.50.  NUMBER OF UNIT SQUARES TC THE NORTH=    5. NUMBER OF UNIT SQUARES TO THE EAST= 6
THE COORDINATES (X.Y) OF THE SOUTHWEST CORNER= ( 687.600.3982.300)

  KNTR     NCOL     NROW    BASIN     X-COORD          Y-COORD      RAIN-GAGE INDEX

    1        1        1      19       687.850         3982.550           0
    2        1        2       1       687.850         3983.050          61
    3        1        3       1       687.850         3983.550          61
    4        1        4       1       687.850         3984.050          61
    5        1        5      19       687.850         3984.550           0
    6        2        1       1       688.350         3982.550          61
    7        2        2       1       688.350         3983.050          61
    8        2        3       1       688.350         3983.550          61
    9        2        4       1       688.350         3984.050          61
   10        2        5       1       688.350         3984.550          61
   11        3        1    1907       688.850         3982.550           0 61
   12        3        2       1       688.850         3983.050          61
   13        3        3       1       688.850         3983.550          61
   14        3        4       1       688.850         3984.050          61
   15        3        5       1       688.850         3984.550          61
   16        4        1      19       689.350         3982.550           0
   17        4        2       1       689.350         3983.050          61
   18        4        3       1       689.350         3983.550          61
   19        4        4       1       689.350         3984.050          61
   20        4        5       1       689.350         3984.550          61
   21        5        1      19       689.850         3982.550           0
   22        5        2       1       689.850         3983.050          61
   23        5        3       1       689.850         3983.550          61
   24        5        4       1       689.850         3984.050          61
   25        5        5       1       689.850         3984.550          61
   26        6        1      19       690.350         3982.550           0
   27        6        2      19       690.350         3983.050           0
   28        6        3       1       690.350         3983.550          61
   29        6        4       1       690.350         3984.050          61
   30        6        5      19       690.350         3984.550           0
```

FIGURE 4-12 (Continued)

```
 19     1     1     1     1    19
  1     1     1     1     1     1
  1     1     1     1     1     1
  1     1     1     1     1    19
 19     1  1907    19    19    19




  0    61    61    61    61     0
 61    61    61    61    61    61
 61    61    61    61    61    61
 61    61    61    61    61     0
  0    61    61     0     0     0
```

```
CATCHMENT   1 IS   22 UNIT SQUARES. THEY ARE:           2   3   4   6   7   8   9  10  12  13  14  15
                                                       20  22  23  24  25  28  29
CATCHMENT   7 IS    1 UNIT SQUARES. THEY ARE:          11
CATCHMENT  19 IS    8 UNIT SQUARES. THEY ARE:           1   5  11  16  21  26  27  30
THE FOLLOWING  1 CATCHMENT ID'S ARE EXAMS WATER-BODIES OR WATER-BODY CELLS:       7
```

```
                               POINT SOURCE INPUT DATA
POINT  NAME       E-RATE1    E-RATE2              X-COORD.  Y-COORD.  STK.HGT   DIAM      VELOC      TEMP-K

  1 L AND M       1.07       1.46E-04             687.600   3984.300  50.0      2.00      3.31       461.
  2 DUR HOS       .147       3.64E-05             689.200   3990.000  11.0      .800      5.20       400.
  3 AMERTOBC      1.73       5.10E-04             688.800   3985.000  61.0      2.40      2.75       446.
```

USE THIS SHEET FOR REPORT TYPING 8-1

TOP

FIGURE 4-12 (Continued)

```
                  AREA SOURCE INPUT AND COMPUTED DATA
NO.    NAME  E-RATE (G/S)
                                                     CORNER LOC. (KM)  WIDTH (KM)  AREA (KM**2)  E-RATE (G/KM**2/S)
  1     C3M         5.18        5.19E-04    685.600  3986.300      2.000        4.000   1.29       1.30E-04
  2     E3R         3.64        3.05E-04    685.600  3984.300      2.000        4.000   .910       7.62E-05
  3     G4R         2.60        1.61E-04    686.600  3982.300      2.000        4.000   .650       4.03E-05
  4     E5C         28.6        3.20F-03    687.600  3984.300      4.000       16.000   1.79       2.00E-04
  5     C1R         1.68        3.35E-05    683.600  3986.300      2.000        4.000   .420       8.37E-05
  6     G6M         5.49        5.63E-04    688.600  3982.300      2.000        4.000   1.37       1.41E-04
  7     E1R         1.68        3.35E-05    683.600  3984.300      2.000        4.000   .420       8.37E-06
  8     H1R         3.78        7.54E-05    683.600  3981.300      3.000        9.000   .420       8.38E-06
  9     K1R         3.78        7.54E-05    683.600  3978.300      3.000        9.000   .420       8.38E-06
 10     G8R         3.06        2.92E-04    690.600  3982.300      2.000        4.000   .765       7.30E-05
 11     B9M         .650        4.03E-05    691.600  3987.300      1.000        1.000   .650       4.03E-05
 12     C9M         1.11        1.04E-04    691.600  3986.300      1.000        1.000   1.11       1.04E-04
 13     D9M         1.39        1.43E-04    691.600  3985.300      1.000        1.000   1.39       1.43E-04
 14     E9M         1.34        1.36E-04    691.600  3984.300      1.000        1.000   1.34       1.36E-04
 15     K4R         6.72        1.34E-04    686.600  3978.300      4.000       16.000   .420       8.38E-06
 16     K8R         6.72        1.34E-04    690.600  3978.300      4.000       16.000   .420       8.38E-06
 17     G10R        1.68        3.35E-05    692.600  3982.300      2.000        4.000   .420       8.37E-06
 18     E10R        1.68        3.35E-05    692.600  3984.300      2.000        4.000   .420       8.37E-06
 19     C10R        1.68        3.35E-05    692.600  3986.300      2.000        4.000   .420       8.37E-06

               CITY SOURCE INPUT AND COMPUTED DATA
NO.  NAME  E-RATE (G/S)
                                               CORNER LOC. (KM)  WIDTH (KM)  CITY (KM**2)  E-RATE (G/KM**2/S)
1 .     RALEIGH      254.       2.65E-04    714.600  3947.300     11.500    132.250  1.92      2.00E-06
      SZ0= 60.00 60.00 50.00 40.00 40.00 30.00

2 POLLUTANTS WILL BE MODELLED IN THIS RUN.
   TOTAL POINT SOURCE EMISSIONS:     2.947        6.924E-04
   TOTAL AREA SOURCE EMISSIONS:      82.43        6.050E-03
   TOTAL CITY SOURCE EMISSIONS:      254.0        2.650E-04
```

USE THIS SHEET FOR REPORT TYPING 8-1/2 x 11

TOP

FIGURE 4-12 (Continued)

```
          DAILY METEOROLOGICAL DATA FOR YEAR: 72 MONTH:  1  DAY NO.:   2
 HOUR   WIND DIR   WIND SPD  P-G STAB  MIX.HGT    AIR TEMP     RAINFALL AMOUNTS

   1     56.000     3.087       4      992.000    279.817             0
   2     72.000     2.572       4      992.000    279.817             0
   3     72.000     1.000       4      992.000    279.817             0
   4    130.000     3.601       4      992.000    279.817             0
   5    126.000     1.000       4      992.000    279.817             0
   6    127.000     1.000       4      992.000    279.261             0
   7    179.000     2.572       4      992.000    279.261             0
   8    246.000     2.058       4      992.000    279.261             0
   9    200.000     3.601       4      992.000    279.261             0
  10    221.000     2.572       4      992.000    280.373             0
  11    265.000     4.116       4      992.000    280.928             0
  12    181.000     4.116       4      992.000    280.928             0
  13    199.000     3.601       4      992.000    282.595             0
  14    217.000     4.116       4      992.000    283.151             0
  15    250.000     5.144       4      992.000    283.706             0
  16    274.000     5.659       4      992.000    284.262             0
  17    248.000     4.630       4      992.000    284.262             1
  18    261.000     3.601       4      992.000    283.151             0
  19    302.000     3.601       5      992.000    281.484             0
  20    265.000     2.572       6      992.000    280.928             0
  21    304.000     2.058       6      992.000    279.817             0
  22    335.000     2.058       6      992.000    279.261             0
  23    317.000     2.572       6      992.000    277.594             0
  24    358.000     2.572       6      992.000    276.483             0
DAILY RAINFALL TOTAL:                                          1
          DAILY METEOROLOGICAL DATA FOR YEAR: 72 MONTH:  1  DAY NO.:   3
 HOUR   WIND DIR   WIND SPD  P-G STAB  MIX.HGT    AIR TEMP     RAINFALL AMOUNTS

   1    340.000     3.087       6      992.000    275.928             0
   2    356.000     2.572       6      992.000    274.817             0
   3      4.000     1.000       6      992.000    273.150             0
   4    340.000     3.087       6      992.000    272.039             0
   5    357.000     3.087       6      992.000    272.594             0
   6     39.000     3.087       6      992.000    271.483             0
   7     16.000     3.087       6      992.000    272.594             0
   8     19.000     1.000       5       74.964    273.705             0
   9     58.000     2.572       4      227.803    275.372             0
  10     64.000     3.601       3      380.642    278.150             0
  11     82.000     4.630       3      533.482    280.928             0
  12     58.000     4.116       3      686.321    282.595             0
  13    111.000     4.630       3      839.160    283.151             0
  14    102.000     4.116       3      992.000    284.262             0
  15     97.000     3.087       3      992.000    284.817             0
  16    110.000     4.116       4      992.000    284.262             0
  17     79.000     2.058       4      992.000    283.151             0
  18     71.000     2.058       5      992.000    279.261             0
  19     67.000     2.058       6      992.000    276.483             0
  20     88.000     2.058       6      992.000    275.372             0
  21     87.000     2.058       6      992.000    278.150             0
  22     89.000     2.058       5      992.000    279.261             0
  23     81.000     3.087       4      992.000    279.817             0
  24     87.000     2.058       4      992.000    279.817             0
DAILY RAINFALL TOTAL:                                          0
```

Reproduced from best available copy.

For Page Numbering

USE THIS SHEET FOR REPORT TYPING 8-1/2 x 11

TOP

FIGURE 4-12 (Continued)

```
          DAILY METEOROLOGICAL DATA FOR YEAR:  72 MONTH:  1   DAY NO.:    4
 HOUR   WIND DIR   WIND SPD   P-G STAB   MIX.HGT    AIR TEMP     RAINFALL AMOUNTS

   1      86.000      1.000      5       992.000    279.817             0
   2     346.000      2.058      4       992.000    278.706             0
   3     358.000      2.058      4       992.000    278.706             0
   4     319.000      3.601      4       992.000    279.261             0
   5      21.000      2.058      4       992.000    279.261             0
   6      62.000      2.572      4       992.000    279.261             0
   7      76.000      1.543      4       992.000    279.261             0
   8     298.000      1.543      4       992.000    279.817             0
   9       6.000      1.543      3       992.000    279.817             0
  10     137.000      2.572      4       992.000    280.373             0
  11     137.000      3.601      4       992.000    281.484             0
  12     233.000      1.029      4       992.000    282.039             4
  13     205.000      2.058      4       992.000    282.595             5
  14     192.000      3.087      4       992.000    285.373             1
  15     211.000      5.144      4       992.000    286.484             1
  16     200.000      4.116      4       992.000    286.484             0
  17     233.000      3.601      4       992.000    286.484             0
  18     209.000      3.601      4       992.000    286.484             0
  19     207.000      5.144      4       992.000    287.040             0
  20     219.000      4.630      4       992.000    287.040             0
  21     217.000      3.601      4       992.000    287.040             0
  22     219.000      3.601      4       992.000    287.595             0
  23     212.000      3.601      4       992.000    287.595             0
  24     188.000      5.659      4       992.000    288.151             0
DAILY RAINFALL TOTAL:                                          11


          DAILY METEOROLOGICAL DATA FOR YEAR:  72 MONTH:  1   DAY NO.:    5
 HOUR   WIND DIR   WIND SPD   P-G STAB   MIX.HGT    AIR TEMP     RAINFALL AMOUNTS

   1     245.000      4.116      4       992.000    288.707             0
   2     234.000      6.173      4       992.000    289.818             0
   3     237.000      4.630      4       992.000    289.818             0
   4     240.000      4.116      4       992.000    289.918             2
   5     242.000      5.144      4       992.000    289.818             0
   6     245.000      6.173      4       992.000    289.818             0
   7     242.000      5.144      4       992.000    289.818             2
   8     245.000      6.689      4       992.000    289.262             1
   9     259.000      4.116      4       992.000    289.262             0
  10     278.000      5.144      4       992.000    288.151             1
  11     248.000      4.116      4       992.000    287.040             2
  12     279.000      6.173      4       992.000    287.595             0
  13     328.000      8.231      4       992.000    283.151             0
  14     320.000      8.231      4       992.000    282.039             0
  15     335.000      7.717      4       992.000    282.595             0
  16     318.000      4.116      4       992.000    282.039             0
  17     329.000      6.173      4       992.000    280.928             0
  18     326.000      4.630      5       992.000    277.594             0
  19     327.000      5.144      5       992.000    277.039             0
  20     327.000      3.087      6       992.000    275.928             0
  21     322.000      4.630      5       992.000    274.817             0
  22     330.000      3.087      6       992.000    273.150             0
  23     338.000      3.087      6       992.000    272.594             0
  24      12.000      3.601      5       992.000    272.594             0
DAILY RAINFALL TOTAL:                                           8
```

Fit Page Numbering

USE THIS SHEET FOR REPORT TYPING 8-1/2 x 11

10 Lines to End of Typing Area

FIGURE 4-12 (Continued)

```
END OF INPUT HOURLY MET. DATA ENCOUNTERED.NORMAL END OF RUN.

      HIGHEST POINT SOURCE HOURLY CONCENTRATION RANKING FOR  POLLUTANT: 1
             RANK: 1 CONCENTRATION:    10.82      (RECEP.SOURCE.HOUR.DAY):     1003    10208
             RANK: 2 CONCENTRATION:    10.22      (RECEP.SOURCE.HOUR.DAY):     1003     8234
             RANK: 3 CONCENTRATION:    9.276      (RECEP.SOURCE.HOUR.DAY):     2003    18178
             RANK: 4 CONCENTRATION:    9.178      (RECEP.SOURCE.HOUR.DAY):     1503    13159
             RANK: 5 CONCENTRATION:    9.178      (RECEP.SOURCE.HOUR.DAY):     1503    15252

      HIGHEST AREA  SOURCE HOURLY CONCENTRATION RANKING FOR  POLLUTANT: 1
             RANK: 1 CONCENTRATION:    449.0      (RECEP.SOURCE.HOUR.DAY):      504    21153
             RANK: 2 CONCENTRATION:    449.0      (RECEP.SOURCE.HOUR.DAY):     1004    21153
             RANK: 3 CONCENTRATION:    449.0      (RECEP.SOURCE.HOUR.DAY):     1504    21153
             RANK: 4 CONCENTRATION:    449.0      (RECEP.SOURCE.HOUR.DAY):     2004    21153
             RANK: 5 CONCENTRATION:    449.0      (RECEP.SOURCE.HOUR.DAY):     3004     5154

      HIGHEST CITY  SOURCE HOURLY CONCENTRATION RANKING FOR  POLLUTANT: 1
             RANK: 1 CONCENTRATION:    2121.      (RECEP.SOURCE.HOUR.DAY):      501     7237
             RANK: 2 CONCENTRATION:    2117.      (RECEP.SOURCE.HOUR.DAY):     1001     7237
             RANK: 3 CONCENTRATION:    2117.      (RECEP.SOURCE.HOUR.DAY):      401     7237
             RANK: 4 CONCENTRATION:    2114.      (RECEP.SOURCE.HOUR.DAY):     1501     7237
             RANK: 5 CONCENTRATION:    2113.      (RECEP.SOURCE.HOUR.DAY):      901     7237

      TEN HIGHEST DAILY AVERAGE CONCENTRATION RANKING FOR   POLLUTANT: 1
             RANK: 1 CONCENTRATION:    790.4      (KATCH.DAY):     7237
             RANK: 2 CONCENTRATION:    777.2      (KATCH.DAY):     1237
             RANK: 3 CONCENTRATION:    768.3      (KATCH.DAY):    19237
             RANK: 4 CONCENTRATION:    570.6      (KATCH.DAY):     7190
             RANK: 5 CONCENTRATION:    548.7      (KATCH.DAY):     7241
             RANK: 6 CONCENTRATION:    543.0      (KATCH.DAY):     1241
             RANK: 7 CONCENTRATION:    539.4      (KATCH.DAY):     1190
             RANK: 8 CONCENTRATION:    536.0      (KATCH.DAY):    19190
             RANK: 9 CONCENTRATION:    535.4      (KATCH.DAY):    19241
             RANK:10 CONCENTRATION:    525.0      (KATCH.DAY):     1225
```

USE THIS SHEET FOR REPORT TYPING 8-1/2 x 11

TOP

FIGURE 4-12 (Continued)

```
HIGHEST POINT SOURCE HOURLY CONCENTRATION RANKING FOR  POLLUTANT: 2
       RANK: 1 CONCENTRATION:    3.122E-03  (RECEP,SOURCE,HOUR,DAY):         1003     10208
       RANK: 2 CONCENTRATION:    2.949E-03  (RECEP,SOURCE,HOUR,DAY):         1003      8234
       RANK: 3 CONCENTRATION:    2.680E-03  (RECEP,SOURCE,HOUR,DAY):         1503     18159
       RANK: 4 CONCENTRATION:    2.680E-03  (RECEP,SOURCE,HOUR,DAY):         1503     15252
       RANK: 5 CONCENTRATION:    2.678E-03  (RECEP,SOURCE,HOUR,DAY):         2003     18178

HIGHEST AREA  SOURCE HOURLY CONCENTRATION RANKING FOR  POLLUTANT: 2
       RANK: 1 CONCENTRATION:    2.389E-02  (RECEP,SOURCE,HOUR,DAY):         1106     21153
       RANK: 2 CONCENTRATION:    2.389E-02  (RECEP,SOURCE,HOUR,DAY):         1206     21153
       RANK: 3 CONCENTRATION:    2.389E-02  (RECEP,SOURCE,HOUR,DAY):         1306     21153
       RANK: 4 CONCENTRATION:    2.389E-02  (RECEP,SOURCE,HOUR,DAY):         1406     21153
       RANK: 5 CONCENTRATION:    2.389E-02  (RECEP,SOURCE,HOUR,DAY):         2906      5154

HIGHEST CITY  SOURCE HOURLY CONCENTRATION RANKING FOR  POLLUTANT: 2
       RANK: 1 CONCENTRATION:    8.501E-04  (RECEP,SOURCE,HOUR,DAY):         2601      7237
       RANK: 2 CONCENTRATION:    8.442E-04  (RECEP,SOURCE,HOUR,DAY):         2101      7237
       RANK: 3 CONCENTRATION:    8.431E-04  (RECEP,SOURCE,HOUR,DAY):         2701      7237
       RANK: 4 CONCENTRATION:    8.383E-04  (RECEP,SOURCE,HOUR,DAY):         1601      7237
       RANK: 5 CONCENTRATION:    8.376E-04  (RECEP,SOURCE,HOUR,DAY):         2801     20225

TEN HIGHEST DAILY AVERAGE CONCENTRATION RANKING FOR   POLLUTANT: 2
       RANK: 1 CONCENTRATION:    1.761E-02  (KATCH,DAY):      7179
       RANK: 2 CONCENTRATION:    1.707E-02  (KATCH,DAY):      7204
       RANK: 3 CONCENTRATION:    1.691E-02  (KATCH,DAY):     19179
       RANK: 4 CONCENTRATION:    1.683E-02  (KATCH,DAY):      7177
       RANK: 5 CONCENTRATION:    1.668E-02  (KATCH,DAY):     19204
       RANK: 6 CONCENTRATION:    1.668E-02  (KATCH,DAY):      7178
       RANK: 7 CONCENTRATION:    1.602E-02  (KATCH,DAY):      1179
       RANK: 8 CONCENTRATION:    1.599E-02  (KATCH,DAY):      7192
       RANK: 9 CONCENTRATION:    1.558E-02  (KATCH,DAY):      7236
       RANK:10 CONCENTRATION:    1.539E-02  (KATCH,DAY):     19177
```

10 Lines to End of Typing Area

For Page Numbering

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA

4.5.1.3 TASK BUILDING AND BATCH EXECUTION COMMANDS FOR DIDOT

The ODL file for DiDOT (DiDOT.ODL) was given in Figure 4-11. Building the DiDOT task could be done in an interactive session such as this:

```
(prompt)        (user response)

PDS             LINK/OPTIONS/TASK/MAP: (/FULL/WIDE)/OVER:DiDOT

OPTIONS?        ACTFIL = 10

OPTIONS?        MAXBUR = 1.024

OPTIONS?        UNITS = 65

OPTIONS?        /
```

Since so many routines are involved, it may be preferable to carry out the LINK as a batch job. Similarly, running the task is presumed to be a batch job, and so the batch control stream might consist of these commands:

```
$JOB        Export                      Run              77777

$RENAME          DB1: DiDOTINP.DAT, FOR005.DAT       (card input data)

$RENAME          DB1: DiDOTMET.DAT, FOR008.DAT       (unformatted meteorological
                                                      data)

$RENAME          DB1: DiDOTPPT1.DAT, FOR061.DAT      (precipitation data, repeat
                                                      for multiple input precipi-
                                                      tation data sets)

$ASSIGN          DB1: 5

$ASSIGN          LPO: 8

$ASSIGN          DB1: 8

$ASSIGN          DB1: 9                  (creating unformatted disk output file)

$ASSIGN          DB1: 61                 (repeat for LUN's 62-65, if needed)

$RUN             DIDOT

$DEASSIGN            5

$DEASSIGN            8

$DEASSIGN            9

$DEASSIGN           61                   (repeat as necessary
```

DROPPED HEAD; BEGIN SECTIONS HERE
BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

```
$RENAME         DB1: FOR005.DAT, DIDOTINP.DAT
$RENAME         DB1: FOR008.DAT, DIDOTMET.DAT
$RENAME         DB1: FOR009.DAT, DIDOTRAW.DAT
$RENAME         DB1: FOR061.DAT, DIDOTPPT1.DAT     (repeat as necessary)
```

file 2

This batch stream should insure that all necessary files are renamed according to the logical unit number references in DiDOT, and assigned to appropriate devices before the task is run, then have the devices deassigned and the original names restored. The DIDOTRAW.DAT files is, of course, the input file to the postprocessor.

file 3

4.5.1.4 TASK BUILDING AND I/O FILE CONSIDERATIONS FOR THE DiDOT POSTPROCESSOR

The DiDOT postprocessor produces the following output:

A) A separate output file for each NPS catchment. Each file contains three types of output records, all of format:

I2, I4, I6, 31E11.4

The three integer fields at the beginning of each record make it possible to sort the records into a coherent order. The first integer field is the catchment number, 1 through 20. The second integer field is the year. The third integer field is a sort flag tailored to the record type. The record types and their contents are:

1. <u>Hourly Wet Flux</u> (tons/acre/hr) on each day rain was recorded, for each pollutant (TSP is pollutant 1). The sort flag (integer field 3) is given by:

   1000* (Day No.) + Pollutant No.,

   so both the day and the pollutant number may be determined from this field (days are numbered 1 to 366). The hourly wet flux values occupy the first 24 of the 31 E11.4 fields.

2. <u>Monthly Average Potency Factors</u> (tons toxicant/tons TSP), the ratio of total toxicant flux for a month to total TSP flux for a month. The sort flag is given by:

   100 + Month No.,

   so the month may be determined from this flag. The potency factor occupies the first of the 31 E11.4 fields.

BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"

EPA-287 (Cin.)
(4-76)

TYPING GUIDE SHEET
BEGIN FIRST LINE OF TEXT HERE
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

3. **Daily TSP Flux** (tons/acre) for each month. From the daily TSP flux and the monthly average potency factor, daily average toxicant flux may be calculated. The sort flag is simply the month number. Records are output for each month of the year, even if some months are absent from the study. Daily flux is set to zero for those months (or days) not in the study period. For each month, if there were N days in the month, then the daily TSP flux values occupy the first N of the 31 E11.4 fields. The Postprocessor is able to determine leap years.

When the records are sorted into numerical order on the first three integer fields (that is, on the first twelve characters of the record), the record order for each year (years in ascending order) is as follows:

D_1

D_2

.

.

.

D_{12}

P_i

P_{i+1}

.

.

.

P_j

$W_{k,1}$

$W_{k,2}$

.

.

.

$W_{1,1}$

$W_{1,2}$

where the D's are daily TSP flux records for months 1 through 12 of the first year, the P's are monthly average potency factor records for months i (the first months of the study) through j (the last month of the study or of the year), and the W's are hourly wet flux records for each rain day, from the year's first rain day (day k) to the year's last rain day (day 1), one record for each of two pollutants. This $D_1, \ldots, D_{12}, P_i \cdots, W_{k,1}, \ldots, W_{1,2}$ cycle is repeated for each year of the study.

6-1/2"
9-1/8"
BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"
PAGE NUMBER

EPA-287 (Cin.)
(4-76)

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

B) A single output file for the EXAMS water body cells, consisting of two types of records, both of format

I1, 10E11.4, I1.

The two record types give dry and wet toxicant fluxes (kg/m^2) per day for the up to ten definable EXAMS cells. The first integer field gives the pollutant number (always 2). The last integer field is used only in the dry flux record, and flags the presence or absence of a wet flux record for the day. If the flag is 1, the immediately following record is that day's wet flux record. If the flag is 0 or blank, the following record is the next day's dry flux record. Flux values are filled in from the left-most of the 10 E11.4 fields, beginning with the lowest numbered EXAMS cells. That is, if catchments 4, 9, and 14 of the study area are EXAMS cells, then the daily flux for catchment 4 will be in the first of the 10 E11.4 fields, flux for catchment 9 will be in the second field, for 14 will be in the third field, and fields 4 through 10 will contain zeroes. The first dry flux record is for the first day of the study period, and the records are consecutive by day, with wet flux records produced on days when rain is recorded.

From the foregoing discussion, plus that of Sections 4.2 and 4.3, the following rules for preparing logical unit number associations should be followed:

9-1/8"

- the single control card and optional potency data matrix are input on unit 15
- the DiDOT output file, referred to as DIDOTRAW.DAT above, is assigned unit number 16
- each catchment identifier specified in the $CATCHS input package to DiDOT will result in a separate output file from the postprocessor. The logical unit number for each is determined by adding 20 to the catchment identifier. Thus, catchment 1 become LUN 21, catchment 11 becomes LUN 31, and catchment 19 becomes LUN 39. By assigning the appropriate LUN to a null device or scratch file, the catchment identifier associated with the marginal area can be eliminated.
- logical unit number 41 is reserved for all the EXAMS cells specified to DiDOT regardless of their individual catchment identifiers. It is important to avoid declaring an LUN to the postprocessor for which the identifier is associated with an EXAMS cell. For example, if identifier 7 was specified as an EXAMS cell to DiDOT through the NEXAMS input in $CATCHS, then LUN 27 should not be declared for the postprocessor. Whether there is one or more than one EXAMS cell in DIDOT, there is only one LUN for them on output from the postprocessor, namely LUN 41.

BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
BEGIN LAST LINE OF TEXT
3/8"

EPA-287 (Cin.)
(4-76)

TYPING GUIDE SHEET

For the task building purposes, the postprocessor does not need to be overlaid. The necessary FORTRAN file names are:

MAIN, INIT, DAYMON, GETPOT, MONCLR, NEWMON, NEWYR,
NUMDAY, PRTPOT, PRTTSP, REC1, REC2, REC3, and YRCLR.

Appropriate options might be:

ACTFIL = 23, MAXBUF = S12, UNITS = 41

This is the maximum number of active files under any circumstances, since it provides for one card input, one printer output, one raw input data set and the maximum number of catchments and EXAMS cells (i.e., a total of 20). At the other extreme, only 4 files may be active, due to only (up to 10) EXAMS cells specified to DiDOT, with no NPSDEP catchments. It may be practical to respecify this parameter and the UNITS parameter and relink the program each time.

The value of 512 for MAXBUF is somewhat too large, as the actual fixed record length is only 353 bytes. This value was chosen, however, because it may improve disk space allocation and utilization. The EXAMS record length is 112 bytes.

4.5.2 NPSDEP

The modified version of NPS runs on the IBM installation at EPA/RTP. It is necessary for the catchment files from the DiDOT postprocessor to be sorted before they can be used by NPSDEP. As shown in Figure 4-2, this can be done on the PDP 11/70 using the SORT command. The specifications are simply to sort the first 12 characters of each fixed-length, 353-byte record in ascending order.

The sorted records must be made available to NPSDEP on the IBM computer. These are input aslogical unit 7. The other files and logical unit number assignments for running NPSDEP were shown in Figure 4-2. Example JCL to run NPSDEP would be similar to this set of commands:

```
//NPSRUN          EXEC          PGM = NPSDEP
//STEPUB          DD            DSNZ(load-module-library), DISP =
                                SHR

//FTO2FOO1        DD            DSNZ(meteorological data[card image]),
                                 DISP = SHR
    (or if provided via in-stream cards,  //FTO2FOO1  DD*)
//FT03F001        DD            DSN=(NPSDEP output file), DISP
                                =(NEW, KEEP),
//                UNIT=(on-line storage), SPACE=(TRK, (5,1), RLSE),
//                DCB=(RECFM=F, LRECL=50, BLKSIZE = 80)
//FT04F001        DD            DSN=(namelist data [card-image],
                                DISP=SHR
   (or, if provided via in-stream cards, //FT04F001 DD *)
```

EPA-287 (Cin.)
(4-76)

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

```
//FT06F001          DD          SYSOUT=A
//FT07F001          DD          DSN=(sorted postprocessor output
                                file), DISP=(OLD, PASS),
//                              DCB=)RECFM=FB, LRECL=353, BLKSIZE
                                =7060)
//*
```

Additional processing steps after NPSDEP executes would be necessary to transmit the output file on unit 3 back to the PDP11/70. By writing them in unblocked format, with a block size of 80 bytes, direct spooling to HASP may be possible by specifying a different SYSOUT class instead of DSN=data-set-name for this file.

4.5.3 NPS/EXAMS Interface and EXAMS

The NPSEXAMS interfacing routine was described in Section 4.2.4. With respect to PDP11/70 job deck setup, the user must assign a logical unit number to each catchment that was modeled by NPSDEP and DiDOT. The LUN is determined by adding 8 to the catchment identifier: catchment 1 becomes LUN 9, 11 becomes 19, 19 becomes 27. Other files used by the interface routine are unit 4, which is an input data set (two cards) containing scaling factors and other program control variables, and units 7 and 8 which are output files. Unit 6 is a dummy or null file.

As with the DiDOT postprocessor, it may be desirable to relink this interface program at each usage, so that the UNITS and ACTFIL options can be set according to the needs of the application. Since the record length of the output files is less than the default maximum value (111 vs. 132), the MAXBUF option need not be specified in linking.

In running EXAMS, logical unit numbers 3 and 4 are null data sets, and units 6 and 8 are printed outputs. (Unit 8 is an error message file.) Units 5 and 7 correspond to the standard system input and reference data input logical units respectively.

Logical units 9 and 10 correspond to the two output files of the interface, units 7 and 8, respectively. Finally, logical unit 11 is associated with the EXAMS output data set from the DiDOT postprocessor, that is, logical unit 41 in that program.

With respect to linking the revised version of EXAMS, the existing version of the overlay description need not be modified. The ODL for EXAMS is shown in Figure 4-13, and the overlay structure is given in Figure 4-14. With respect to link options, ACTFIL-10, and UNITS=11 should suffice. Since the interface output files and the EXAMS input file from the postprocessor have record lengths less than 132, MAXBUF need not be specified.

12 of 13

BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
BEGIN LAST LINE OF TEXT
3/8"

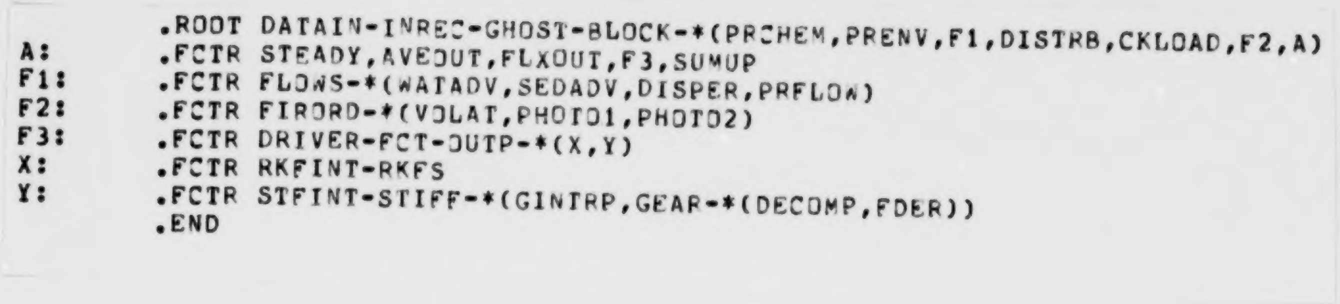
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

114

USE THIS SHEET FOR REPORT TYPING 8-1/2 x 11

TOP

10 Lines to End of Typing Area



```
          .ROOT DATAIN-INREC-GHOST-BLOCK-*(PRCHEM,PRENV,F1,DISTRB,CKLOAD,F2,A)
A:        .FCTR STEADY,AVEOUT,FLXOUT,F3,SUMUP
F1:       .FCTR FLOWS-*(WATADV,SEDADV,DISPER,PRFLOW)
F2:       .FCTR FIRORD-*(VOLAT,PHOTO1,PHOTO2)
F3:       .FCTR DRIVER-FCT-OUTP-*(X,Y)
X:        .FCTR RKFINT-RKFS
Y:        .FCTR STFINT-STIFF-*(GINTRP,GEAR-*(DECOMP,FDER))
          .END
```

FIGURE 4-13: OVERLAY DESCRIPTION OF EXAMS

4-12

For Page Numbering

USE THIS SHEET FOR REPORT TYPING 8-1/2 x 11

TOP

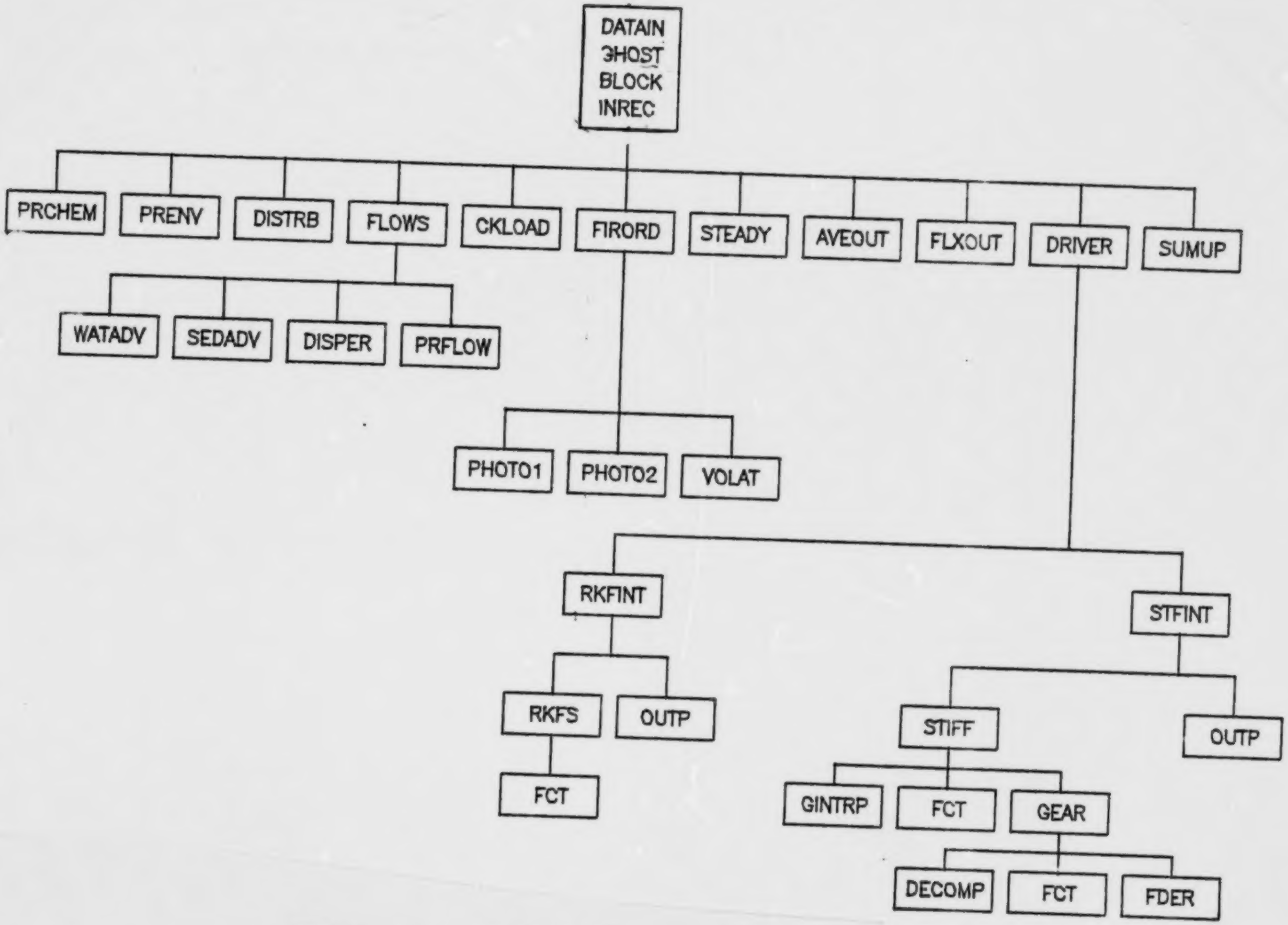

DATAIN
GHOST
BLOCK
INREC
PRCHEM
PRENV
DISTRB
FLOWS
CKLOAD
FIRORD
STEADY
AVEOUT
FLXOUT
DRIVER
SUMUP
WATADV
SEDADV
DISPER
PRFLOW
PHOTO1
PHOTO2
VOLAT
RKFINT
STFINT
RKFS
OUTP
FCT
STIFF
OUTP
GINTRP
FCT
GEAR
DECOMP
FCT
FDER

FIGURE 4-14: OVERLAY STRUCTURE OF EXAMS

4-13

10 Lines to End of Typing Area

For Page Numbering

5.0 SENSITIVITY ANALYSIS

In order to demonstrate the utility of ALWAS, as well as to evaluate its response to several key input parameters, a sensitivity analysis has been performed. The analysis is designed to fully test the model's capabilities, and to indicate the sensitivity of the model's response across reasonable ranges of uncertainty in the model input parameters.

Benzo(a)Pyrene (BaP) was selected as the toxic pollutant for the study for a number of reasons. BaP is a carcinogenic pollutant with a relatively complete data base including important environmental properties and atmospheric emission rates for various source categories nationwide. Furthermore, the major loading of BaP to surface waters of the U.S. is from direct deposition and runoff, and the major contributor to BaP in runoff is from atmospheric deposition (Risk Assessment PAH's, ADL, 1981). This is true for relatively few toxic pollutants, so BaP is an ideal pollutant for testing of ALWAS.

The Third Fork Creek watershed of Durham, North Carolina, was selected as the site for the sensitivity analysis, primarily because of the availability of a calibrated NPS input data set. However, the study area is also suitable for a test of ALWAS because it is urban, with both point and area sources of TSP and BaP. The nearby city of Raleigh is ideally suited as a "city-sized" area source, which tests another capability of ALWAS.

The analysis is principally designed to determine the effect on surface water quality of uncertainty in input parameters which control the delivery of pollutants to the EXAMS water body. The parameters which were tested are the settling velocity, deposition velocity, and precipitation scavenging ratio of BaP (DiDOT parameters), and the exponents in the sediment washoff equations (NPSDEP parameter). The deposition parameters for BaP have been estimated using the procedures of section 4.4. As demonstrated by Tucker and Preston (1982), these estimation procedures are valid to within roughly one order of magnitude, so it is appropriate to vary these parameters by roughly an order or magnitude for sensitivity analysis. We have departed from the standard approach to sensitivity testing since we have not independently varied each parameter. Independent variation of each parameter is not appropriate, since they are not independent: each depends on the vapor/aerosol partitioning and particle size. Consequently, certain variations in settling velocity are not possible with related changes in deposition velocity and scavenging ratio. The combination of values used in the tests are physically realizable. The deposition parameters for TSP have also been evaluated, using Section 4.4 method, as well as available field data. The deposition parameters for TSP are considered more reliable and have not been varied in sensitivity testing. The exponents in the sediment washoff equations of NPSDEP are varied from 1.0 to 2.6 consistent with their possible range

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD: BEGIN SECTIONS HERE

(Donigian and Crawford, 1976), while the best estimate value (1.8) results from NPS calibration to the Third Fork Creek watershed. Parameter values for various sensitivity tests are shown in Table 5-1.

Estimates of multi-media source emissions of BaP are designed to be realistic in order of magnitude, but do not accurately represent conditions in the watershed. Source strengths are based on typical source strengths of urban activities, such as those occurring in Durham, but are not based on an extensive local analysis of BaP sources.

In order to show the full capability of ALWAS to simulate multi-media impacts, the sensitivity test case includes a source of groundwater contamination, as well as the deposition of air emissions. Using a screening procedure which accounts for typical seepage rates from waste lagoons and soil/water partitioning, Falco, et al. (1980) estimated the unit release rate of BaP to surface waters from waste lagoons at .008 mg/m^2 yr. Assuming that a waste lagoon, receiving boiler cleaning wastes containing BaP, was situated in the watershed near the EXAMS water body, we obtained an estimate of groundwater (interflow) BaP loading of 3.6×10^{-9} kg/hr. In order to emphasize BaP impacts, the Third Fork Creek watershed drains into a small pond in the sensitivity analysis simulations. This pond, like the waste lagoon described in the previous paragraphs, is completely hypothetical.

5.1 PREPARATION OF INPUTS FOR SENSITIVITY ANALYSIS

5.1.1 DiDOT

5.1.1.1 Source/Receptor Inputs

Section 5.1, explains the preparation of inputs to DiDOT for a sensitivity analysis of the model. Preparation of area, point, and cities sources; receptor grid and model parameters for the Third Fork Creek Basin in Durham, North Carolina, will be discussed. The generation of the rainfall file and the binary meteorological data will not be discussed in this section, but can be found in section 5.1.1.2.

Description of the Study Area: As discussed by Donigian and Crawford (1976), the Third Fork Creek basin is located in the eastern section of the North Carolina Piedmont, a region of gently rolling hills. The stream flows in a southerly direction and is tributary to the New Hope, Haw, and Cape Fear River systems. Upper Third Fork Creek, simulated in this study, drains an area of 433 hectares (1,069 acres) located within the city limits of Durham, North Carolina. The drainage basin is primarily composed of two shallow valleys with relatively minor flood plains along the lower reaches of the streams. Surface runoff generally follows natural drainage paths with a few man-made channels. No storm sewer system exists; storm runoff occurs largely as surface runoff in street gutters, small pipes, and culverts under roads.

The region experiences a moderate climate without distinct wet and dry seasons. Summer storms tend to be brief, high-intensity thunderstorms, and convective showers. Winter and spring storms are of longer duration and lower intensity and

BEGIN LAST LINE OF TEXT
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
3/8"
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

TYPING GUIDE SHEET
BEGIN FIRST LINE OF TEXT HERE
DROPPED HEAD; BEGIN SECTIONS HERE
CENTER OF PAGE
TOP OF IMAGE AREA
9-1/8"
6-1/2"
BEGIN LAST LINE OF TEXT
3/8"
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

TABLE 5-1

INPUT PARAMETER VALUES FOR SENSITIVITY TESTING

Run Number	W_s(BaP) m/sec	W_d(BaP) m/sec	r(BaP)	W_s(TSP) m/sec	W_d(TSP) m/sec	r(TSP)	JSER	JEIM
1	$1x10^{-4}$	$7.5x10^{-3}$	$6x10^4$	$4x10^{-3}$	0.02	$5x10^5$	1.8	1.8
2	0.01	0.015	$6x10^5$					
3	0.1	0.1	$3x10^6$					
4	$1x10^{-4}$	$7.5x10^{-4}$	$6x10^4$					
5		$7.5x10^{-3}$	$6x10^3$					
6			$6x10^5$					
7			$6x10^4$				1.0	1.0
8							2.6	2.6

are caused by migratory low-pressure weather systems. Mean annual precipitation is approximately 1,200 millimeters (47 inches) of which less than 25 millimeters (1 inch water equivalent) occurs as snowfall with little significant accumulation.

The Third Fork Creek basin, shown in Figure 5-1, represents a typical urbanized area in the Piedmont region of the Southeastern United States. The basin encompasses a variety of land uses, including:

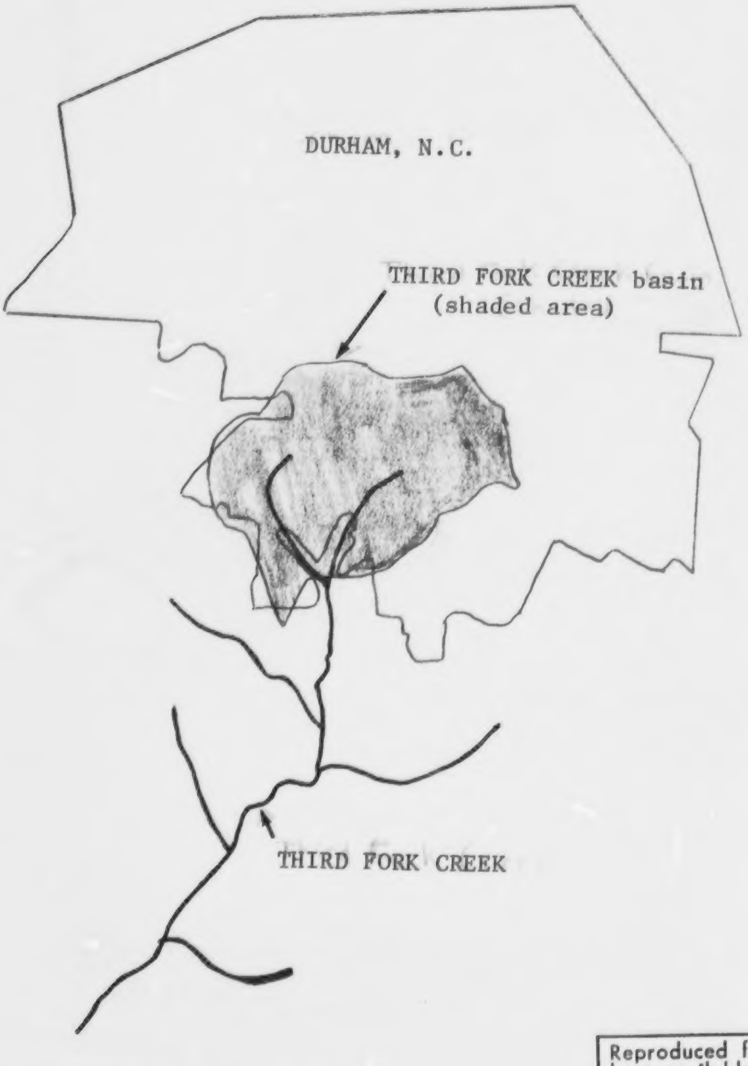
- high- and low-density housing units of varying quality,
- undeveloped land,
- shopping centers,
- portion of the central business district,
- institutional buildings (churches and schools) among scattered, small businesses,
- an urban redevelopment section,
- a tobacco processing plant,
- a completed section of expressway,
- a cemetery,
- slums,
- railroad yard, and
- a flood plain utilized mainly as a city park.

Air pollution emissions from areas other than the city of Durham will affect the study area. DiDOT can process emissions from nearby cities and emissions from the surrounding countryside up to four area source grids from the study area.

These data requirements made it necessary to inventory emissions for all of Durham County and for the city of Raleigh, which is the second closest city to the study area.

Point Sources: Three point sources for the city of Durham were identified from data obtained from the state. All but one source (Durham County Hospital) is located within the city limits. A TSP emission rate was available for these sources, but a BaP emission rate was not. All the point sources were residual oil burning boilers. Benzo(a)Pyrene (BaP) emissions from these boilers were estimated by finding the ratio of nationwide BaP emissions to nationwide hydrocarbons (HC) emissions from residual oil burning boilers and multiplying the ratio by the point source HC emission rate. An example is shown below.

Figure 5-1
THIRD FORK CREEK BASIN


DURHAM, N.C.
THIRD FORK CREEK basin
(shaded area)
THIRD FORK CREEK

Reproduced from
best available copy.

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD, BEGIN SECTIONS HERE

$$\frac{\text{Nationwide BaP}^{a}}{\text{Nationwide HC}^{b}} \quad \frac{19 \text{ tons/year}^{a}}{15{,}039 \text{ tons/year}^{b}} \times \text{Source HC } 3 \text{ tons/hr} = 3.79\times10^{-3} \text{ BaP emission rate ton/year}$$

(a) Source: Nagda, N.L., 1979, Environmental Carcinogens and Human Cancer; Estimation of Exposure to Carcinogens in Ambient Air. RTP, NC.

(b) Source: NEDS 1975 Emission Inventory.

BaP emission rates were estimated in this manner for all point sources.

Three point sources used in the sensitivity analysis are Durham County Hospital and two boilers associated with tobacco processing operations. The point source characteristics can be found in Table 5-2.

Area Sources: As mentioned before, the area source grid extends four grids in every direction from the study area. The grid is 11 km by 10 km and contains 110 units that make up 19 area sources. A diagram of the area source grid is found in Figure 5-2.

6-1/2"

An "R" in a grid square designates a square which is predominantly rural and a "C" designates an urban square. A blank grid square means that it contains both rural and urban areas. Area sources are named using the numbering scheme at the top, the lettering scheme along the side, and the type of grid it is, i.e., rural (R), urban (C), and blank or mixed (M). For example, grid designated by B-3 would be B3M, and the large urban area source block would be designated by E5C because that grid is in the southwest corner of the area source block.

Emission rates for all the rural grids are the same and from all the city grids are the same. Parts of the grid have mixed (rural and urban) emission rates, that were included in the larger source blocks to consolidate area sources. Blank or mixed grids contain emissions based on what percentage of the grid resides in rural and urban areas.

Table 5-3 contains emissions from transportation-related sources, and Table 5-4 emissions from miscellaneous source. Both tables of emissions are for Durham County in 1975 (source: Dept. of Natural Resources and Community Development, Division of Environmental Management: Raleigh, N.C.).

As can be seen from these tables, an asterisk denotes sources that can be considered urban. These urban sources were multiplied by the ratio of population in the city of Durham to the population in Durham County to estimate what percentage of emissions could be alloted to an urban grid and a rural grid. An example for motor vehicle emissions is shown below.

$$\frac{\text{Pop. Durham City}^{a}}{\text{Pop. Durham Co.}^{a}} \quad \frac{101{,}224}{132{,}681} \times 493 \text{ tons/yr.} = 376 \text{ tons/yr. (urban)}$$

$$493 \text{ tons/yr. (total)} - 376 \text{ tons/yr. (urban)} = 117 \text{ tons/yr. (rural)}$$

(a) Source: County and City Data Book 1977; U.S. Dept. of Commerce.

BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
BEGIN LAST LINE OF TEXT
3/8"

PAGE NUMBER
EPA-287 (Cin.)
(4-76)

TABLE 5-2

POINT SOURCE CHARACTERISTICS

SOURCE NAME	EMISSION RATES (g/s) TSP	EMISSION RATES (g/s) BaP* (estimated)	Stack Height (M)	Stack Diam (M)	Exit Vel (M/S)	Stack Temp (^{o}k)	Vol. Flux (M^3/S)	utm Coordinates (km) X	utm Coordinates (km) Y
Tobacco Processing A	1.07	1.46×10^{-4}	50.	2.		461.	10.4	687.6	3984.3
Durham County Hospital	.14	3.64×10^{-5}	11.	.8	5.2	400.		689.2	3990.0
Tobacco Processing B	1.73	5.10×10^{-4}	61.	2.4		446.	12.5	688.8	3985.0

*BaP emissions were estimated via the ratio of national BaP emissions in residual oil boilers to the ratio of national hydrocarbon emissions for the same class of sources. These emission rates are used for illustrative purposes in the sensitivity analysis only, not as an accurate estimate of BaP emissions.

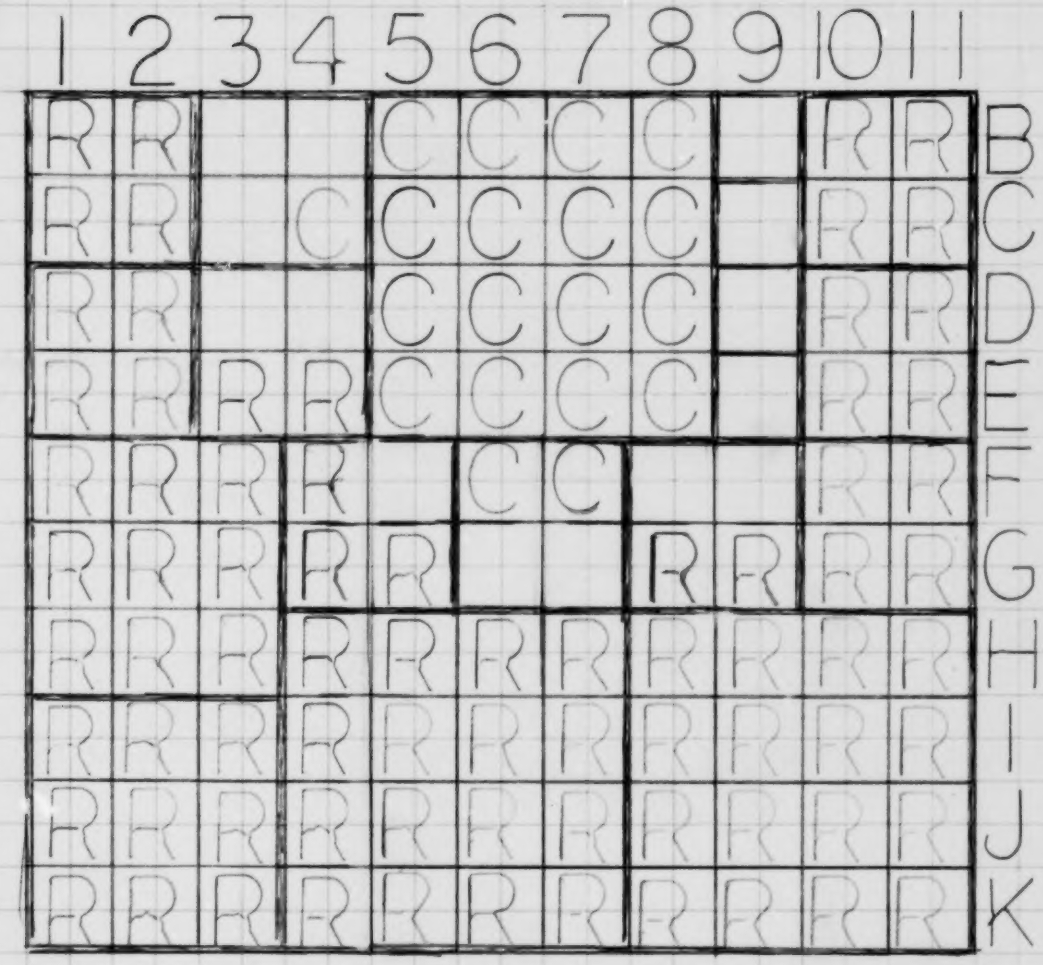
EPA-287 (Cin.)
(4-76)

PAGE NUMBER

BEGIN
FIRST
TYPING GUIDE SHEET
TOP OF

FIGURE 5-2

AREA SOURCE GRID*



	1	2	3	4	5	6	7	8	9	10	11
B	R	R			C	C	C	C		R	R
C	R	R		C	C	C	C	C		R	R
D	R	R			C	C	C	C		R	R
E	R	R	R	R	C	C	C	C		R	R
F	R	R	R	R		C	C			R	R
G	R	R	R	R	R			R	R	R	R
H	R	R	R	R	R	R	R	R	R	R	R
I	R	R	R	R	R	R	R	R	R	R	R
J	R	R	R	R	R	R	R	R	R	R	R
K	R	R	R	R	R	R	R	R	R	R	R

*110 grid squares

Reproduced from best available copy.

*19 area sources: area sources denoted by heavy lines

3/8"
AND ILLUS TRATIONS

PAGE NUMBER
EPA-287 (Cin.)
(4-76)

TABLE 5-3

EMISSIONS FROM TRANSPORTATION - RELATED SOURCE

DURHAM COUNTY 1975

Source	POLLUTANT (tons/yr) TSP	HC
Motor Vehicles*	493	3656
Off-Highway Fuel* Usage	32	367
Railroads*	29	108
Vessels		8
Aircraft		1
Fugitive Dust		
Paved Roads*	5615	
Unpaved Roads	10,668	
Total	16,857	4140

*Sources associated with urban areas.

BEGIN LAST LINE OF TEXT
3/8''
OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

TABLE 5-4

EMISSIONS FROM MISCELLANEOUS SOURCES

DURHAM COUNTY 1975

Source	POLLUTANT (tons/yr)	
	TSP	HC
Wind Erosion	21	
Agricultural Tilling	316	
Construction Activity*	88	
Agricultural & Land Clearing Burning	14	15
Structural Fires	255	66
Slash Burning	2	2
Forest Fires	3	4
Miscellaneous * Anthropogenic Sources	47	308
Total	746	395

*Sources associated with urban areas

LAST LINE OF TEXT
3/8"
TABLES AND ILLUSTRATIONS
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

TYPING GUIDE SHEET
BEGIN FIRST LINE OF TEXT HERE
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD; BEGIN SECTIONS HERE

This operation was repeated for all urban type sources.

All non-asterisk sources were considered rural. Rural emission rate estimates varied depending on source type. For example, wind erosion and unpaved road emissions were estimated by multiplying their emissions by the ratio of land area in the city of Durham to the area of Durham County. An example for unpaved roads is shown below:

$$\frac{\text{Durham City}^{a}}{\text{Durham County}^{a}} \quad \frac{39.7 \text{ sq. mi}}{295.7 \text{ sq. mi}} \times 10{,}668 \text{ tons/yr.} = 1440 \text{ tons/yr. (urban)}$$

$$10{,}668 \text{ tons/yr. (total)} - 1440 \text{ tons/yr. (urban)} = 9228 \text{ tons/yr. (rural)}$$

(a) Source: County and City Data Book 1977; U.S. Dept. of Commerce

All rural sources with the exception of wind erosion, vessels, and unpaved roads are unique only to rural squares. In other words, one would not expect to find slash burning or forest fires in an urban environment. For this reason, the emissions of the following sources were alloted entirely to a rural square: agricultural tilling; agricultural and land clearing burning; aircraft; slash burning; and forest fires. Aircraft emissions were entirely alloted to rural squares because the airport is not located within the city of Durham.

BaP emissions were estimated in approximately the same manner as TSP emissions. For example, the ratio of nationwide BaP from motor vehicles, to nationwide HC from motor vehicles, was multiplied by the HC motor vehicle emission rates for both the urban and rural squares to estimate BaP emission rates. This operation was done for all the source types and was summed to estimate urban and rural BaP emissions. It should be noted that the ratio of nationwide BaP emissions to HC emissions from residential furnaces was the largest. Because of this, BaP emissions from miscellaneous anthropogenic sources (MAS) were the largest. BaP emissions from the other sources were at least, and in some cases more, an order of magnitude smaller. This large difference causes BaP emissions from other sources to be insignificant compared to MAS. The references for nationwide BaP and HC emissions for this analysis are the same as noted in Section 5.1.3.

Table 5-5 summarized TSP, HC, and BaP emission rates estimated for the area source grids. As can be seen from this table, BaP emission rates are very small.

A complete listing of the area source inventory and an illustration of the area source grid in relation to the study area are shown in Figures 4-3 and 4-4 .

City Source: For the study area, there is only one city source, the city of Raleigh which is approximately 20 miles southeast of the study area. Emission rates were estimated by multiplying urban BaP and TSP emission rates for Durham by the ratio of the population of the city of Raleigh to the population of the city of Durham. An example for TSP is shown below.

$$\frac{\text{Pop. City of Raleigh}^{a}}{\text{Pop. City of Durham}^{a}} \quad \frac{134{,}231}{101{,}224} \times 6447 \text{ tons/yr.} = 8550 \text{ tons/yr.}$$

BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
BEGIN LAST LINE OF TEXT
3/8"

EPA-287 (Cin.)
(4-76)

TABLE 5-5

SUMMARY OF EMISSIONS USED AS INPUT TO DIDOT

POLLUTANT
(tons/yr.)

	TSP	HC	BaP* (estimated)
Rural	11,136	1092	2.22×10^{-1}
Urban	6,447	3443	7.14×10^{-1}
Total	17,583	4535	9.36×10^{-1}

g/s-km^2 for INPUT to DiDOT

	TSP	BaP * (estimated)
Rural	0.42	8.38×10^{-6}
Urban	1.80	2.00×10^{-4}

---

*These emission rates are used for illustrative purposes in the sensitivity analysis, not as accurate estimates of BaP emissions.

BEGIN LAST LINE OF TEXT
3/8''
OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
PAGE NUMBER
EPA-287 (Cin.)
(4-76)

FIGURE 5-3

AREA SOURCE EMISSION INVENTORY

Source Name	TSP Emission (g/s)	BaP Emission (g/s)	SW COORDINATES (km) X	SW COORDINATES (km) Y	Side Length (km)
$ARSRCE01 C3M	5.176	5.19E-04	685.6	3986.3	2.
$ARSRCE02 E3R	3.64	3.05E-04	685.6	3984.3	2.
$ARSRCE03 G4R	2.6	1.61E-04	686.6	3982.3	2.
$ARSRCE04 E5C	28.57	3.20E-03	687.6	3984.3	4.
$ARSRCE05 C1R	1.68	3.35E-05	683.6	3986.3	2.
$ARSRCE06 G6M	5.49	5.63E-04	688.6	3982.3	2.
$ARSRCE07 E1R	1.68	3.35E-05	683.6	3984.3	2.
$ARSRCE08 H1R	3.78	7.54E-05	683.6	3981.3	3.
$ARSRCE09 K1R	3.78	7.54E-05	683.6	3978.3	3.
$ARSRCE10 G8R	3.06	2.92E-04	690.6	3982.3	2.
$ARSRCE11 B9M	.65	4.03E-05	691.6	3987.3	1.
$ARSRCE12 C9M	1.11	1.04E-04	691.6	3986.3	1.
$ARSRCE13 D9M	1.39	1.43E-04	691.6	3985.3	1.
$ARSRCE14 E9M	1.34	1.36E-04	691.6	3984.3	1.
$ARSRCE15 K4R	6.72	1.34E-04	686.6	3978.3	4.
$ARSRCE16 K8R	6.72	1.34E-04	690.6	3978.3	4.
$ARSRCE17 G10R	1.68	3.35E-05	692.6	3982.3	2.
$ARSRCE18 E10R	1.68	3.35E-05	692.6	3984.3	2.
$ARSRCE19 C10R	1.68	3.35E-05	692.6	3986.3	2.

CITY SOURCE EMISSION INVENTORY

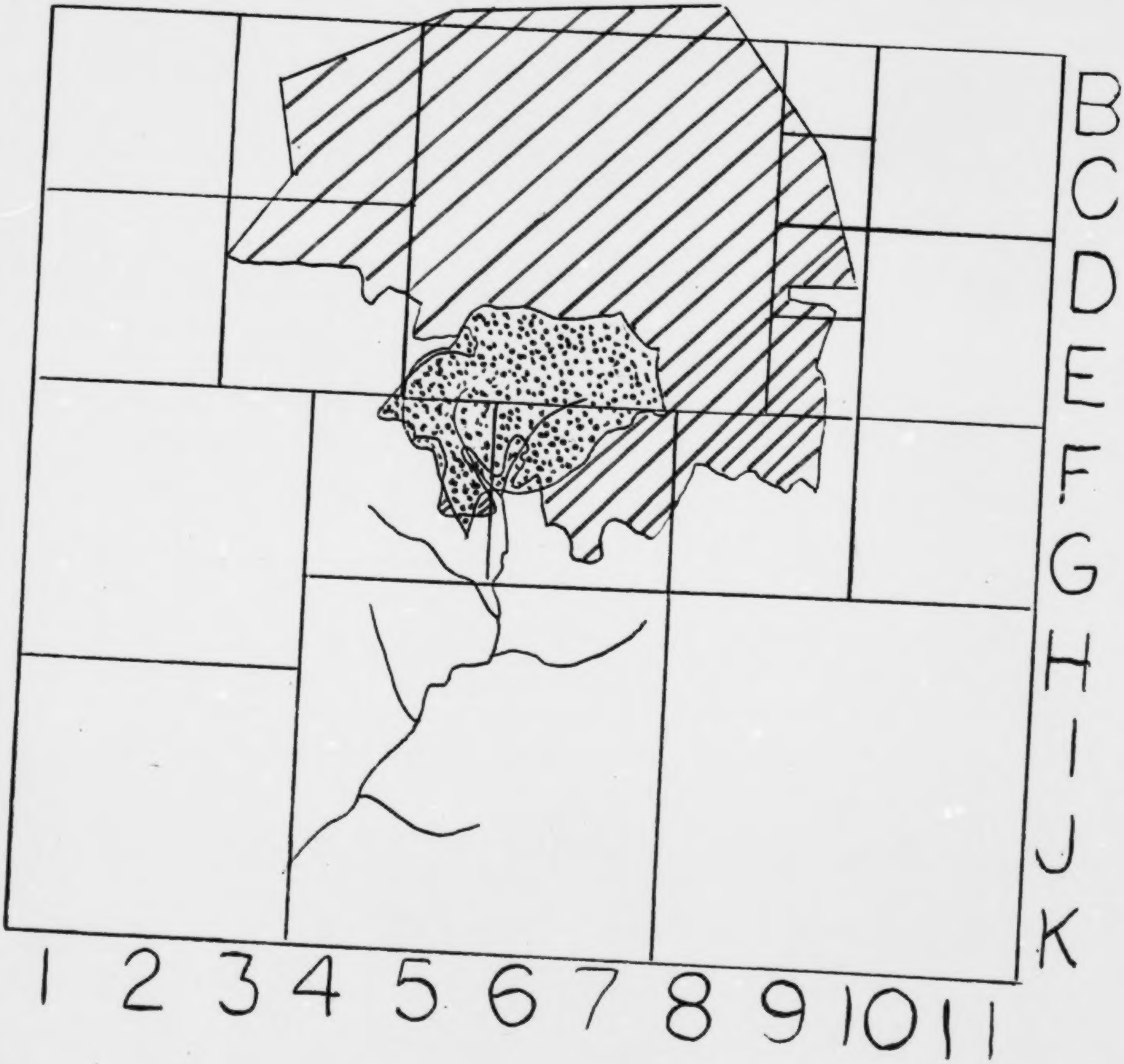
$CITY 01 RALEIGH	254.	2.65E-04	714.6	3947.3	11.

Reproduced from best available copy.

FIGURE 4-4

Fig 5-1

AREA SOURCE GRID
IN RELATION TO STUDY AREA


B
C
D
E
F
G
H
I
J
K
1
2
3
4
5
6
7
8
9
10
11

The city of Raleigh has a land area of 125 km^2 so it is assumed to be an area source 11 km on a side. The coordinates of the southwest corner are:

X coordinate 714.4 km
Y coordinate 3947.1 km

(a) Source: County and City Data Book 1977; U.S. Dept. of Commerce.

Receptor Grid: The receptor grid is 3.0 km by 2.5 km and was designed to extend to the outermost points of the study area. The grid spacing is 0.5 km and extends north 2.5 km and east 3.0 km from the southwest corner. The utm coordinates of the southwest corner are:

X coordinate 687.6
Y coordinate 3982.3

An illustration of the receptor grid overlaid on the study area is contained in Figure 5-5. Each receptor is numbered consecutively from bottom to top starting at the southwest corner. Catchments are assigned to the receptor grid in that manner. Catchment assignment is discussed in detail in the DiDOT documentation.

Notice in Figure 5-5 the small pond where Third Fork Creek exits the study area. The pond is hypothetical only: no pond actually exists at this location. It was included in the sensitivity analysis test to emphasize the BaP accumulation in surface water bodies, and the precise characteristics of the pond were chosen for convenience to be identical to the POND ecosystem currently included as a representative ecosystem for EXAMS analysis.

5.1.1.2 METEOROLOGICAL DATA

Data from the National Weather Service Station at the Raleigh-Durham airport were obtained from the National Climatic Center. These data were processed by the CRSMET preprocessor for subsequent input to DiDOT. Precipitation data from the NCC tape were not used. Instead, the precipitation record from a gage on the Third Fork Creek watershed, used in NPS calibration by Donigian and Crawford (1976) was used, after integrating the 15 minute data over a onehour averaging time. Note that the treatment of precipitation data is consistent with requirements that DiDOT input precipitation data be identical to the NPSDEP input and use a one-hour averaging time.

Results

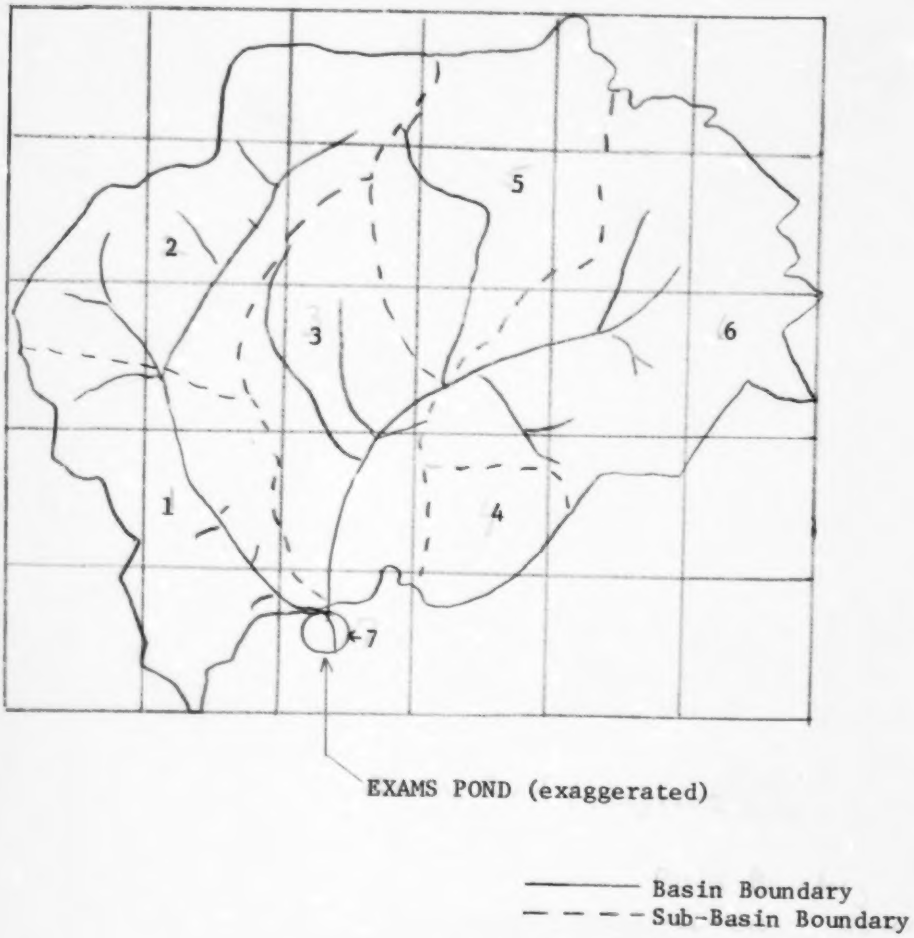
Figure 5-6 shows resultant water quality in the hypothetical pond draining Third Fork Creek Watershed. Concentration values presented are the dissolved BaP in the water column (water) and the adsorbed concentration in the bottom sediments (sediment). The two "spikes" in the water concentration output are caused by major rainstorms which carried significant amounts of accumulated BaP to the creek.

EPA-287 (Cin.)
(4-76)

BEGIN FIRST
TYPING GUIDE SHEET
CENTER
TOP OF

FIGURE 5-5

RECEPTOR GRID OVERLAID ON
THIRD FORK CREEK STUDY AREA*

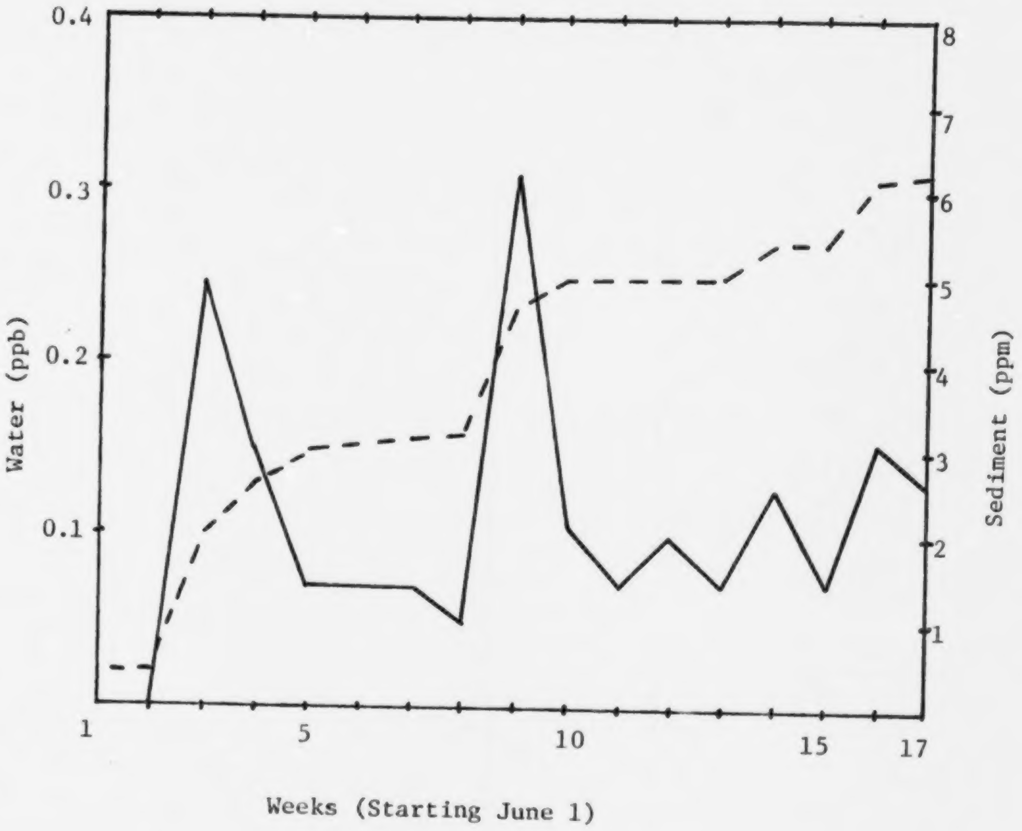

5
2
3
6
1
4
7
EXAMS POND (exaggerated)
——— Basin Boundary
– – – Sub-Basin Boundary

*Numbers in Sub-Basins indicate catchment

BEGIN FIRST LINE OF TEXT HERE
TYPING GUIDE SHEET
CENTER OF PAGE
TOP OF IMAGE AREA
DROPPED HEAD: BEGIN SECTIONS HERE

FIGURE 5-6

BaP CONCENTRATION IN HYPOTHETICAL POND: SENSITIVITY RUN 1



BEGIN LAST LINE OF TEXT
3/8"
BOTTOM OF IMAGE AREA; OUTSIDE DIMENSION FOR TABLES AND ILLUSTRATIONS
PAC

EPA-287 (Cin.)
(4-76)

TYPING GUIDE SHEET

Run #1 incorporates the best estimate for each of the parameters (assuming the mass median diameter of airborne BaP is 0.75 μm), while subsequent runs incorporate plausible variations from those best estimates reflecting the range of uncertainty inherent in estimated deposition parameters. Runs #2 and #3 reflect "best estimates" for the deposition parameters if the mass median diameter were 10 or 20 μm, respectively. Both wet and dry deposition parameters are affected by a change in the particle size. Run #4 reduces the dry deposition velocity by an order of magnitude while all other parameters remain unchanged. Runs #5 and #6 reflect order of magnitude changes in the precipitation scavenging ratio, while all other parameters are the same as Run #1. In runs #7 and #8 two parameters which affect sediment washoff from the land surface in NPSDEP are varied to the limits of the possible range (Donigian & Crawford, 1976) for comparison with run #1 in which the value calibrated for Third Fork Creek watershed was used. Runs #7 and #8 will allow the comparison of uncertainties in the land segment of the model with uncertainties in the air deposition algorithms. Comparison of results of each of the sensitivity runs is made on the basis of the final water column dissolved concentration, which was 0.13 ppb for run #1.

Figure 5-7 illustrates the results of these tests. First, the affect of variation in the mass median diameter is dominant (Figure 5-7 a) since particle size affects both wet and dry deposition parameters. The range tested (0.8-20 μm) is much broader than the actual uncertainty inherent in the estimation of the mass median diameter of airborne BaP. However, in considering the full set of airborne toxic substances, mass median diameters may vary over this range.

Figures 5-7 b and 5-7 c demonstrate that variability of an order of magnitude in dry depostion velocity and precipitation scavenging ratio result in less than proportionate changes in resultant water quality. The resultant water quality during the simulation period was slightly more sensitive to changes in scavenging ratio than to dry deposition velocity. This greater sensitivity to the wet deposition parameter must be viewed in the context of the relatively high rainfall rate during the simulation period, averaging 5.5 in/mo. In more arid climates, the dry deposition velocity would be a relatively more important parameter.

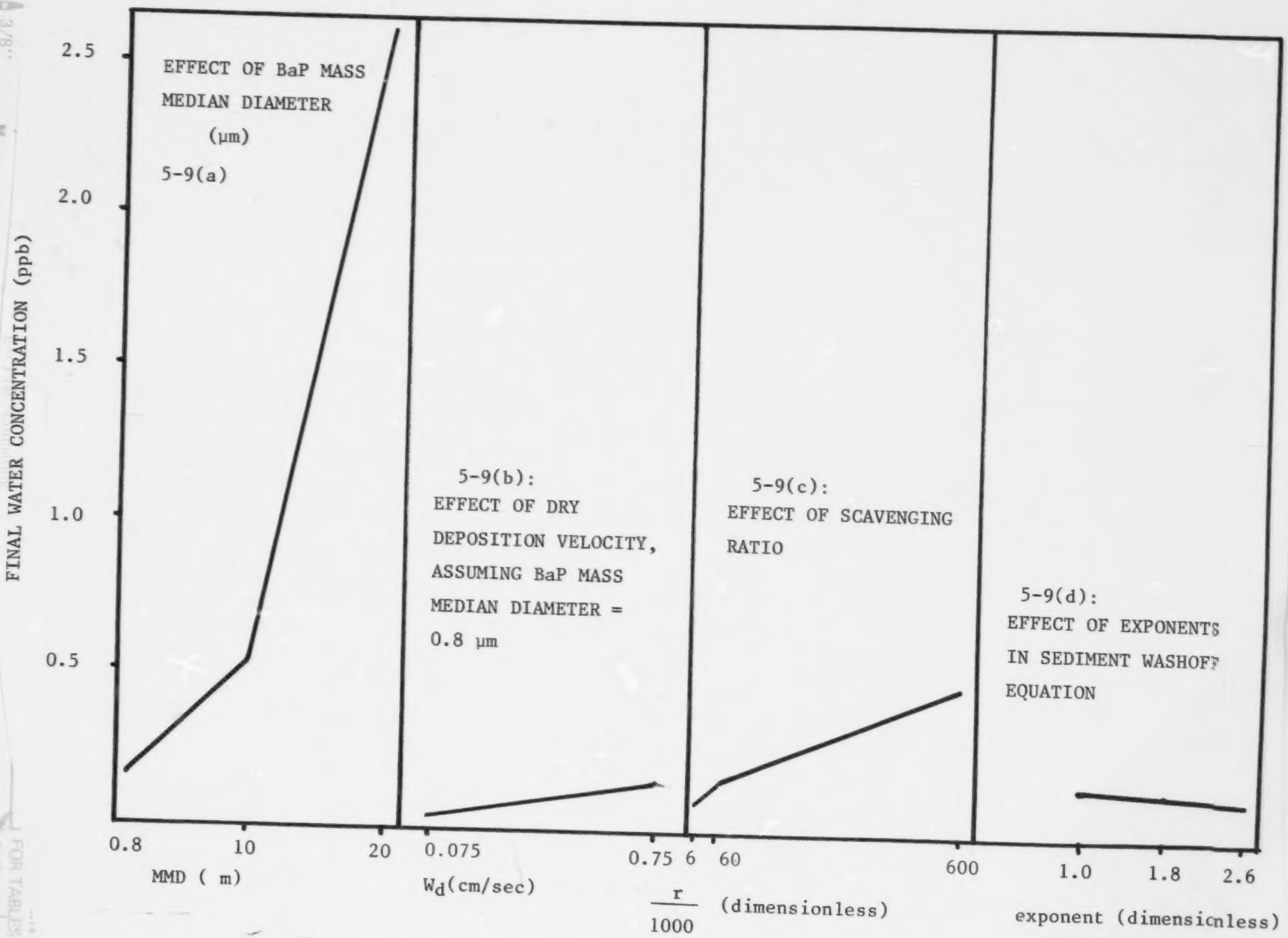
Finally, Figure 5-7 d demonstrates that the resultant water quality is relatively insensitive to the exponents in the sediment washoff equation. This occurs for a number of reasons, all acting jointly in the test simulation:

- essentially all BaP in runoff comes from atmospheric deposition
- the watershed is calibrated, thus many sensitive parameters of the land segment are fixed
- deposited contaminants are held near the soil surface and ALWAS algorithms cause sediment particles so contaminanted to be washed off first

Given these conditions, it is clear that ALWAS water quality results are much more sensitive to uncertainties in the deposition parameters than to uncertainties in sediment washoff.

EPA-287 (Cin.)
(4-76)

FIGURE 5-7: RESULTS OF SENSITIVITY ANALYSIS



EPA-287 (Cin.)
(4-76)

REFERENCES

1. Aravamudan, K., M. Bonazountas, A. Eschenroeder, D. Gilbert, L. Nelken, K. Scow, R. Thomas, W. Tucker, C. Unger. 1980. An Environmental Partitioning Model for Risk Assessment of Chemicals: Part I. Narrative Description of the Methodology, prepared by Arthur D. Little, Inc., for U.S. EPA, Office of Toxic Substances. Contract 68-01-3857.

2. Arthur D. Little, Inc. 1981. Air-to-Surface Pathway Evaluation Methodology, draft final report prepared for U.S. EPA, Office of Water Regulation and Standard, Monitoring and Data Support Division, under Task V of Contract 68-01-5949.

3. Arthur D. Little, Inc. 1981. Risk Assessment of Polynuclear Aromatic Hydrocarbons, draft final report prepared for U.S. EPA, Office of Water Regulation and Standards under Contract 68-01-5949.

4. ASME. 1968. Recommended Guide for the Prediction of the Dispersion of Airborne Effluents. M. Smith (ed.), New York.

5. Bigg, F., G. Ayers, and D. Turvey. 1978. Measurement of the Dispersion of a Smoke Plume at Large Distances from the Source, Atmos. Environ., 12, 1815-8.

6. Briggs, G. S. 1970. Some Recent Analysis of Plume Rise Observation, Second Intntl. Clean Air Congress, Intntl. Union of Air Pollution Prevention Assoc., Washington, D.C.

7. Briggs, G.S. 1973. Chimney Plumes in Neutral and Stable Surroundings, Atmos. Environ., 6, 507-10.

8. Burns, L.A., D.M. Cline, and R.R. Lassiter. 1980. EXAMS, An Exposure Analysis Modeling System, U.S. EPA, Athens Environmental Research Laboratory.

9. Businger, J.A., J.C. Wyngaard, Y. Igumi, and E.F. Bradley. 1971. Flux-profile Relationships in the Atmospheric Surface Layer. J. Atmos. Sci., 28, 181-9.

10. Cowherd, C., Jr., K. Axetell, Jr., C.M. Guenther, and G.A. Jutze. 1974. Development of Emission Factors for Fugitive Dust Sources. Midwest Research Inst., Kansas City, MO. EPA-450/3-74-037.

11. Culkowski, W.M. and M.R. Patterson. 1976. A Comprehensive Atmospheric Transport and Diffision Model. Oak Ridge National Laboratory, ORNL/NSF/EATC-17.

12. Donigian, A.S. and N.H. Crawford. 1976. Modeling Nonpoint Pollution From the Land Surface, EPA-600/3-76-083.

13. Dumbauld, R.K., J.E. Rafferty, and H.E. Cramer. 1976. Dispersion-deposition from Aerial Spray Releases, presented at Third Symposium on Atmospheric Turbulence, Diffusion and Air Quality, Raleigh, N.C., Oct. 19-22, 1976.

14. Eisenreich, S.J., et al. 1981. Airborne Organic Contaminants in the Great Lake Ecosystem, Environ. Sci. Tech., 15, 30-8.

15. Ermak, D. 1977. An Analytical Model for Air Pollutant Transport and Deposition from a Point Source. Atmos. Environ., 11, 231-7.

16. Falco, J.W., L.A. Mulkey, R.R. Swank, Jr., R.E. Lipscei, and S.M. Brown. 1980. A Screening Procedure for Assessing the Transport and Degradation Solid Waste Constituents in Subsurface and Surface Waters.

17. Gatz, D.F. 1976. Wet Deposition Estimation Using Scavenging Ratios, J. Great Lakes Res., Supp. 1 to Volume 2, 1976.

18. Gifford, F.A. 1978. Atmospheric Transport and Dispersion over Cities, Nuclear Safety, 13, 391-402.

19. Hill, A.C. 1971. Vegetation: A Sink for Atmospheric Pollutants, J. Air Pollution Control Assoc., 21, 341-6.

20. Hill, A.C., and E.M. Chamberlain. 1974. The Removal of Water Soluble Gases from the Atmosphere by Vegetation, in Atmosphere-Surface Exchange of Particulate and Gaseous Pollutants, Engelmann, R.J. and G.A. Sehmel (eds.).

21. Horst, T.W. 1977. A Surface Depletion Model for Deposition from a Gaussian Plume. Atmos. Environ., 11, 41-6.

22. Junge, C.E. 1977. Basic Considerations about Trace Constituents in the Atmosphere as Related to the Fate of Global Pollutants, in Fate of Pollutants in the Air and Water Environments, I.H. Suffet (ed.), Wiley and Sons, New York.

23. Kraus, E.B. 1972. Atmosphere-Ocean Interaction, Clarendon Press, Oxford.

24. Lassiter, R.R., G.L. Baughman, and L.A. Burns. 1978. Fate of Organic Substances in the Aquatic Environment in State-of-the-Art in Ecological Modeling, v. 7, proceedings of Conf. on Ecological Modeling, Copenhagen, Denmark, 28 Aug. - 2 Sept., 1978, S.F. Jorgensen (ed.), Intntl. Society for Ecological Modeling, Copenhagen, Denmark, pp. 219-248.

25. Lawson, T.J. and S. Uk. 1979. The Influence of Wind Turbulence, Crop Characteristics and Flying Height on the Dispersal of Aerial Sprays, Atmos. Environ., 13, 711-5.

26. Lucas, D. 1958. The Atmospheric Pollution of Cities, International J. of Air Pollution, 1, 71-86.

27. Maybank, J., K. Yoshida, P. Grover. 1978. Spray Drift from Agricultural Applications. J. Air Pollution Control Assoc., 28, 1009-14.

28. Nagda, N.L. 1979. Environmental Carcinogens and Human Cancer; Estimation of Exposure to Carcinogens in Ambient Air. EPA, Research Triangle Park.

29. NEDS. 1975. National Emissions Report, National Emissions Data System of the Aerometric and Emissions Reporting System. U.S. EPA, Research Triangle Park.

30. NTIS Publication No. PB-284-488. 1977. Compilation of Air Pollutant Emission Factors.

31. Pasquill, F. 1061. The Estimation of Dispersion of Windborne Material, Meteorol. Mag., 90, 33.

32. Pasquill, F. 1974. Atmospheric Diffusion. John Wiley and Sons, New York, pp. 429.

33. Peterson, E.W. 1977. Wet and Dry Deposition - A Synopsis Containing Estimates of Deposition Velocities for Some Pollutant and Trace Gases in the Atmosphere. EPA, Corvallis Environmental Research Laboratory, CFRL-037.

34. Sehmel, G.A., and W.H. Hodgson. 1974. Predicted Dry Deposition Velocities, for Presentation at the Atmosphere-Surface Exchange of Particulate and Gaseous Pollutants, Symposium, Richland, Washington, September 4-6. BNWL-5125.

35. Seinfeld, J.H. 1975. Air Pollution: Physical and Chemical Fundamentals, McGraw-Hill, New York.

36. Slade, D.H. (ed.) 1968. Meteorology and Atomic Energy, U.S. Atomic Energy Commission.

37. Slinn, W.G.N, *et al.* 1978. Some Aspects of the Transfer of Atmospheric Trace Constituents Past the Air Sea Interface. Atmos. Environ., 12, 2055-87.

38. Smith, J.H. W.R. Mabey, N. Bohonos, B.R. Holt, S.S. Lee, T.W. Chou, D.C. Bomberger, and T. Mill. 1977. Environmental Pathways of Selected Chemicals in Freshwater Systems. Part II: Laboratory Studies. EPA-600/7-78-074.

39. Sutton, O. 1953. Micrometeorology, McGraw-Hill, New York.

40. Turner, D.B. 1967. Workbook of Atmospheric Dispersion Estimates. HEW, Public Health Service Publication No. 999-AP-26.

41. Tucker, W.A., W.J. Lyman and A.L. Preston. 1982. Estimation of the Dry Deposition Velocity and Scavenging Ratio for Organic Chemicals, presented at 4th International Conference on Precipitation Scavenging, Dry Deposition and Resuspension, November 29-December 3, 1982, Santa Monica, California.

42. Tucker, W.A. and A.L. Preston. 1983. Procedures for Estimating Atmospheric Deposition Properties of Organic Chemicals, Water, Air and Soil Pollution (in press).

43. Woodruff, N.P., F.N. Siddoway, and D.W. Fryrear. 1972. How to Control Wind Erosion. Agricultural Research Service, Department of Agriculture, Bulletin 354.

APPENDIX A

REVIEW AND EVALUATION OF MODELS FOR THE ATMOSPHERIC DISPERSION, DEGRADATION, AND DEPOSITION OF TOXIC POLLUTANTS

1.1 INTRODUCTION AND BACKGROUND

The special purpose multimedia model called for in this project makes use of specified simulation algorithms for the soil, runoff, and water portions of the description of chemical movement through the environment. The air algorithm, however, is left unspecified and designated as a task for formulation. In undertaking this task, we made maximum use of available techniques for describing air concentrations in terms of emission and meteorological variables. As will be seen in the discussions below, these models have been traditionally developed for the purpose of establishing emissions limitations necessary to achieve compliance with ambient standards.

A second aspect of the formulation of the air portion of the multimedia model is the coupling between the air and the surface; via water or soil. The state of the art in this area is far less developed than that of pure air quality models because the phenomenology described has not entered into the environmental management process in the past. The recent interest in long-range transport modeling and in acid rain, have led to research emphasis on air-to-surface transfer processes.

This report presents the background for single-medium air models presently available in the literature and then describes a representative set of specific models including their input requirements, their structure, and their output. Following the discussion of data requirements for air quality models, the coupling requirements posed by the Non Point Source (NPS) and the Exposure Analysis Modeling System (EXAMS) are outlined. Not only will the mathematical requirements of the coupling be covered, but also the questions of consistency, accuracy, and resolution. The main sections of the report describe the specific model formulations to be used in the multimedia simulation along with the appropriate transfer mechanisms and provide a narrative explanation of computational procedure and anticipated problems.

1.2 TRADITIONAL OBJECTIVES OF AIR QUALITY MODELS*

Planners and decision makers have required reliable methods of predicting air quality improvements in terms of emission reduction in order to control air pollution by regulation of mobile and stationary sources. Any methodology for relating contaminant concentration to the emission sources under specified meteorological conditions is termed an air quality model. These models may vary from simple arithmetic exercises to complex computer simulations thereby providing many paths to the needed answers. Although short-term deadlines have often restricted enforcement agencies to use only the most rudimentary models, the availability of emissions and meteorological data in certain selected regions has encouraged the use of relatively detailed computer models.

---

*The material in Subsections 1.2 and 1.3 is a summary of extensive review material already covered in the literature.[1] The interested reader is referred to that for pre-1976 information and to Subsection 1.4 for more recent model reviews.

The choice of the model that is used has traditionally depended upon the immediate objectives. For example, one must consider who will be using the model and what kind of questions it would be expected to answer. The accessibility of the model from the standpoint of either practical hand calculations or computer implementation will influence the choice. Another consideration is the extent to which the model has been tested for reliability against an adequate data base.

Historically, scientific purposes were fundamental to the first applications of methods for predicting air quality because of the need to understand the results of field measurement programs or of routine monitoring data. This exposure to the real world emphasized both the strengths and weaknesses of various modeling schemes and formed the first step in an iterative process involving the redesign of models and the planning of new field measurement programs. Thus, models were used for designing the field experiments or monitoring systems, and the measured data from the field provided the basis for improvement of the modeling art.

Regulatory applications also provided motivation for the development of reliable models. Long-term air quality projections are implicit in any scheme designed to improve the environment through specific controls on primary pollutant emissions. Thus, modeling found application both in the establishment of the legislation and in the subsequent regulations. The implementation planning process mandated by the Clean Air Act Amendments of 1970 is largely based on modeling results that indicate the alternatives needed to achieve attainment of the ambient standards. Under these plans, the approval of specific projects under either non-attainment or prevention-of-significant-deterioration rules specifically involves model applications.

Systems have been proposed that use models as the central logic in a real-time control system that directs regulation of emission sources based on near-term predicted meteorological conditions. Although intermittent control systems such as this were explicitly prohibited by the 1977 amendments to the Clean Air Act, this concept has been utilized for the control of sources throughout urban areas in certain prefectures in Japan. For these real-time warning systems or control systems, the air quality model provides the logic in the feedback loop that links air quality improvements to emissions reductions under specific forecast meteorological conditions. Clearly, the resolution and reliability for the air quality model in the real-time application both must be of higher quality than those required in the long-term applications outlined above.

1.3 TECHNICAL APPROACHES

1.3.1 Model Classification

In attempting to categorize air quality models, one must probe in many directions. One such direction is whether the model is statistical or deterministic in nature. Statistical models depend on extensive measured data and utilize patterns that exist exclusively among observable quantities. Deterministic models attempt to use mechanistic principles incorporating cause in their structure through mathematical relationships. However, most deterministic models contain varying degrees of empiricism. Few, if any, models, for example, utilize turbulent diffusion formulations based on first principles, but rather employ measured values of dispersion parameters. The same is true with regard to the atmospheric chemistry formulations.

One of the cautions frequently expressed is that if too many disposable parameters exist in deterministic models, they indeed become statistical models in disguise; for all they accomplish is a very elegant version of curve fitting.

Another classification of model relates to the time scale of interest. Ambient air quality standards are stated for measurement averaging periods ranging from one hour to one year. However, for computational purposes, reducing the time scale below one hour for a typical resolution cell size in a model is often necessary. Spatial scales of interest vary from meters up to hundreds of kilometers depending on whether the object of interest is a roadway or an entire region of the nation containing urban and rural areas. Large spatial scales are also indicated when multi-day simulations become necessary for even a moderately sized urban area. Under certain weather conditions, recirculations can cause interactions of the pollution levels of one day with those of the next day. Typical resolution specifications couple the spatial scales through the temporal scales because of the range of transport times available in the atmosphere. Therefore, we are not faced with the overwelming job of filling out the entire matrix of time scales and space scales, but rather we must proceed along some broad diagonal relating the quantities.

Perhaps one of the most fundamental methods of classifying models is to categorize them by methodology. Statistical or rollback type models place great emphasis on historical data through their use of empirical formulas. The rollback models embody the principle that reductions in emissions are reflected by improvements in air qualityf following a straight line, a curved line, or some complex surface that expresses a direct relationship.* These models work best for cases in which the geographical and temporal distribution of the emitters is not changed. The straight line version can apply only to the pollutants that do not undergo chemical transformations in the atmosphere. Thus, an example scenario with a porportional linear rollback applied is a rapidly instituted retrofit control on vehicle emissions enclosed on the entire area simultaneously. Corrections for irregularity and distribution caused by non-uniform growth may be applied using dispersion techniques. The great advantages of the linear roll-back model, of course, are its use of aggregated emissions statistics and its mathematical simplicity. Improved versions of the rollback to account for chemical change have been suggested by using either air quality data or smog chamber data.

---

*See for example, (References 2-12, 20, 42, 43)

Dispersion models may take the form of the simple box formulation or a non-uniform description following a Gaussian formula.* Dispersion models based on the box formulation envision a control volume fixed relative to the surface through which air circulates and into which pollutants are emitted. Either uniform mixing or duct-flow may be assumed in order to get a mean or peak concentration, respectively. A horizontally projected area of the box is most often taken to be the area approximated by the emission sources, and the height of the box, to be some average value of atmospheric mixing depth. Variants of the box model include the moving box formulation that envisions an air parcel moving with the wind being represented by a box of polluted air. In this formulation the height of the box is again assumed to be the mixing depth, however, in this case it may be variable. Also, in the case of the moving box model the horizontal spatial resolution is more often taken to be the resolution of the emissions inventory; i.e., a much smaller cell size than the entire source-complex area.

Gaussian dispersion models are based on approximate solutions to the species transport equation for point or area sources under the influence of a steady wind field. For non-reactive pollutants or for pollutants undergoing a first-order reaction, it is possible to superimpose the solutions of multiple Gaussian plumes in order to determine the effect of an array of sources on a single receptor point. Computationally, this is convenient because it allows sequential consideration of each source element thereby highlighting its influence. Time-dependent formulations use a Gaussian puff approach that removes the limitation of the steady windfield assumption from the calculation and discretely treats puffs as they spread according to the Gaussian law. These models are illustrated by specific formulations in the next sub-section.

Species-conservation models utilize finite difference techniques combining the effects of emission transport, and, in some cases, chemical reactions.** This class of models is the most elaborate simulation currently applied to air quality analysis. both time and space are subdivided into cells in these models so that the accumulated effect of emissions, transport, and meteorological variations are included. While this class of model offers the maximum in detail of description, it often goes beyond the available data base with its input requirements. The use of a synthetic data base with this class of model may delude the user into a high level of confidence by producing large volumes of output data that offer only limited advantages over simpler models. Clearly, where sufficient detail exists in the input data base and in the validation test base, the finite difference models are potentially far better than the others in fidelity of description.

---

*See for example, References 13 through 17.

**See for example, References 18 through 29.

1.3.2 Phenomenology of Air-to-Surface Transfer Processes

As implied in an earlier subsection, this aspect of the development is in a far more limited state than the air quality models themselves. Air-to-surface transfer processes divide naturally into two classes: (1) pollutant transfer to condensed phases in the air that subsequently deposit on the surface, and (2) direct deposition to the surface itself through a variety of processes.

The first class of phenomenology occurs when a gaseous or particulate pollutant associates itself with rainfall. Gaseous pollutants may be transferred to raindrops by diffusion as the raindrops fall through the air and be dissolved or reacted in solution within the raindrop thereby transporting to the surface. This is one of the washout processes. Washout can also occur when particulate pollutants are entrained by falling raindrops much in the way a scrubber removes particles from a gas stream. Rainout of a particular pollutant occurs when the droplet forms, using the particle as a nucleus. A second stage in the rainout process is introduced when a gaseous pollutant condenses on the surface of a pre-existing aerosol particle and then becomes involved in the nucleation of raindrops.

The direct air-to-surface transfer processes fall in the category termed "dry deposition." In this class of phenomenology, either the aerosol particle bearing the condensed phase pollutant on its surface impacts and sticks to the surface, or the pollutant molecule itself finds its way to the surface and sticks. The likelihood of sticking to the surface is expressed mathematically through a net deposition velocity or through a reflection coefficient expressed at the surface. In these cases, the air-to-surface transfer rate may be controlled predominantly by the reaction with or adherence to surface material, or by the rate of diffusion through the surface boundary layer or its associated laminary sublayer. Although expressions are given in later sections for the calculations of deposition velocity, the interactions between chemistry and diffusion in surface processes for specific materials is still only partially understood. For practical fate analyses, it is appropriate to express extreme behavior either as diffusion-limited (with perfect accomodations at the surface) or reaction-limited (with some moderate value of sticking efficiency specified).

1.4 REQUIREMENTS FOR COUPLING NPS AND EXAMS

The emission pattern description set by the coupling requirements to NPS and EXAMS of the air quality model is established by the range of influence established by both emission source strength and dispersion conditions. This means that sources too weak or too distant to contribute measurable transfers from air-to-surface should be neglected. Also, the spatial accuracy of the emission source location description is set by the resolution of the soil and water models in the horizontal plain. EXAMS has no specific horizontal resolution cell size except that set by the total size of water body or implied size of compartment chosen. NPS, on the other hand, has catchment areas of an irregular nature that establish a horizontal cell size. To achieve consistency among air, soil, and water models, the emission source description into the air should not require the user to specify air emission sources on any size grid smaller than that employed for the runoff model, NPS.

The temporal resolution of the emission source description is almost completely determined by the needs of the air model itself because the time resolution of the soil and water models ranges from at least hours to days while that of the air model generally involves minutes to hours. Chemical specification of the air emissions need only be limited to a single dominant species because NPS has no explicit chemistry at all, and EXAMS treats only the transport and degradation of a single organic species.

The coordinate system constraints on the air model are dictated by the resolution cell network needed by NPS. This is best accomplished by developing an air model that produces concentration at the surface or transfer rates to the surface as arbitrary functions of the horizontal coordinates. These time-dependent descriptions of concentration exposures and transfers to the surface can then be converted into appropriate integrated quantities that serve as inputs to the surface model once their coordinate system requirements have been defined. This will involve the formulation of geometry algorithms to coordinate the output of the air model (which is sensitive to the wind direction) with the surface-fixed characteristics of the NPS and EXAMS models. The details of the concentration and transfer rate formulation along with the procedures for computation in terms of geometrical compatibility have been described.

1.5 REVIEW OF AVAILABLE MODELS - DATA SHEET SUMMARY

The following pages illustrate in summary form the classes of models referred to above. In each case, the model names are given, the basic reference is supplied and the methodology is briefly described. Mathematical structure of the model is addressed by stating the source geometry, transfer removal processes and the basic equations. The implementation of each model is characterized by the output, the validation data, and the physical limitations of application.

MODEL: Air Resources Laboratory Model (ARL)

BASIC REFERENCE: Heffler, J.L., Taylor, A.D. and Ferber, G.J. "A regional-continental scale transport, diffusion and deposition model." NOAA Tech. memo. ERL ARL-50, June 1975.

METHODOLOGY:

The ARL model is a hybrid model in which the advection process is simulated by a trajectory model. The model is designed for regional scale applications at distances ranging from 100 km to 1000 km.

The trajectory model may be used to calculate trajectories for a month, a season or a year from any origin in the Northern Hemisphere at 6-hour intervals, from 1 to 10 days in duration and forward or backward in time. Terrain heights used as input to the model are obtained from data tapes containing average heights in one-degree squares.

For short-term analysis (12 to 24 hours), the concentrations are obtained using a Gaussian puff model. The instantaneous puff is transported along trajectories and diffused simultaneously.

SOURCE GEOMETRY:

The source is assumed to be an instantaneous puff. A continuous source is represented by a series of puffs and interpolating them linearly.

TRANSFER AND REMOVAL MECHANISMS:

The pollutant is assumed to be chemically inactive. The deposition is permitted by both dry and wet removal processes.

BASIC EQUATIONS:

The diffusion of the puff is given by:

$$C_m = \left[Q / 2\pi \sigma_h^2 z_m \right] \exp\left(- R^2 / 2\sigma_h^2 \right)$$

C_m = Air concentration in the mixed layer
Q = Emission output per puff
σ_h = Horizontal standard deviation
R = Distance from the source
z_m = Height of the mixed layer

The dry deposition is given by:

$$C_d = \frac{C_m V_d \Delta t}{z_m}$$

V_d = Is the deposition velocity
Δt = Is time interval

The wet deposition is calculated by:

$$C_w = \frac{C_m E P \Delta t}{z_p}$$

E = Empiorically averaged scavenging ratio
P = Precipitation rate
z_p = Precipitation layer height

OUTPUT:

Time-averaged concentration at various receptor locations.

VALIDATION:

The model has been validated between 50 km and 150 km.

LIMITATIONS:

The model cannot be used at short distances from the sources.

MODEL: Atmospheric Turbulence and Diffusion Laboratory Model (ATDL).

BASIC REFERENCE: Hanna S.R., "Description of the ATDL computer model for dispersion from multiple sources." in Industrial Air Pollution Control, Ann Arbor Science, Ann Arbor, MI, 1973, pp 23-32.

METHODOLOGY:

ATDL model is a simple one-box model which may be useful for a preliminary assessment The concentration of a non-reactive chemical substance is assumed to be directly proportional to source strength and inversely proportional to wind speed. Stability is taken into account by an empirical factor.

SOURCE GEOMETRY:

The source is always assumed to be an area source. The effect of point sources can be included in the formulation by considering them separately using the Gaussian Plume formula.

TRANSFER AND REMOVAL MECHANISMS: None

BASIC EQUATIONS:

The simple ATDL model uses the following equation to determine the concentration:

$$\chi = \frac{CQ}{\bar{u}}$$

χ = Ground level air concentration
u = Wind speed
Q = Area source emissions
C = dimensionless parameter;
50 unstable atmospheric conditions
200 neutral atmospheric conditions
600 stable atmospheric conditions

In a slightly more complex version of ATDL mode, advection from upwind grid squares is also considered:

$$\chi = \sqrt{\frac{2}{\pi}}\left[\left(\frac{\Delta x}{2}\right)^{1-b} \Big/ \left(\underset{c}{\cancel{a}}\,u\,(1-\overset{d}{\cancel{b}})\right)\right] \left[Q_0 + \sum_{i=1}^{N} Q_i\left\{(2i+1)^{1-\cancel{b}\,d} - (2i-1)^{1-\overset{d}{\cancel{b}}}\right\}\right]$$

Δx = grid size (m)
N = number of upwind squares being considered
i = indicator of the particular upwind square
Q_0 = emission rate of grid square in which receptor is located
Q_i = emission rate of i^{th} upwind grid square

$$\left.\begin{array}{l} c = 0.05 \\ d = 0.75 \end{array}\right\} \text{average values for neutral conditins}$$

OUTPUT:

Average concentration at receptor points.

VALIDATION:

A recent validation study by Miller (1) indicates that the complex form of ATDL model predicts the concentrations better.

LIMITATIONS:

- Point sources are to be treated separately.
- The model is not applicable where removal mechanisms are significant.

REFERENCES:

(1) Miller, C.W., "An Application of the ATDL Simple Dispersion Model," J. Air Pollut. Control. Assoc., 28, 8, pp. 798-800, August 1978.

MODEL: APRAC and APRAC - 1A

BASIC REFERENCE:

Mancuso, R.L. and F.L. Ludwig. "User's Manual for the APRAC-1A Urban Diffusion Model Computer Program." Publication No. EPA-650/3-73-001 (NTIS Pb 213091), Environmental Protection Agency, Research Triangle Park, NC 27711, September 1972.

A Practical Multipurpose Urban Diffusion Model for Carbon Monoxide (NTIS accession number PB 196-003).

Field Study for Initial Evaluation of an Urban Diffusion Model for Carbon Monoxide (NTIS accession number PB 203-469).

Evaluation of the APRAC-1A Urban Diffusion Model for Carbon Monoxide (NTIS accession number PB 210-813).

Walter F. Dabberdt, F.L. Ludwig, and Warren B. Johnson, Jr., "Validation and Applications of an Urban Diffusion Model for Vehicular Pollutants," Atmos. Environ. 7:603 (1973).

W.B. Johnson, F.L. Ludwig, W.F. Dabberdt, and R.J. Allen, "An Urban Diffusion Simulation Model for Carbon Monoxide," J. Air Pollut. Control Assoc. 23: 490 (1973).

METHODOLOGY:

APRAC is a model which computes hourly average carbon monoxide concentrations for any urban location. The model calculates contributions from dispersion on various scales: extraurban, mainly from sources upwind of the city of interest; intraurban, from freeway, arterial, and feeder street sources; and local, from dispersion within a street canyon. APRAC requires an extensive traffic inventory for the city of interest.

SOURCE GEOMETRY:

User-specified set of traffic links (line sources). All receptors and sources are at ground level.

TRANSFER AND REMOVAL MECHANISMS: None

BASIC EQUATIONS:

Extraurban -
$$\chi_e = \frac{5.15 \times 10^{-11} F}{uL}$$

F = annual fuel consumption within 22.5° sector extending from 32 km to 1,000 km upwind of receptor.

Intraurban -
$$\chi_{ij} = \frac{0.8 Q_i}{u a_{ij}} \quad \frac{x_{i+1}^{1-b_{ij}} - x_i^{1-b_{ij}}}{1-b_{ij}}$$

Until this expression equals the "box model value"
$$\frac{Q_i}{uL}\left(x_{i+1} - x_i\right)$$

Thereafter the box model formula is used.

i = upwind area segment label
u = stability class label
a_{ij} and b_{ij} from $(Q_z)_{ij} = a_{ij}\, x^{b_{ij}}$ for x within segment j

Street Canyon - Lee side
$$\chi_L = \frac{kQ_s}{[u+0.5]\left([x^2+z^2]^{1/2}+L_0\right)}$$

Windward side
$$\chi_W = \frac{kQ_s(H-z)}{(u+0.5)WH}$$

Intermediate wind direction (less than ± 30° from street (direction) is given by:
$$\chi_I = \frac{1}{2}\left(\chi_L + \chi_W\right)$$

where X = horizontal distance from traffic lane (m)
z = height above pavement (m)
K = constant = 7
L_o = vehicle size = 2m
u = rooftop wind speed (m/s)
Q = CO emission rate (g/s-m)
W = street width (m)
H = average building height - 38.8m

OUTPUT:

Hourly concentration and frequency distribution at each receptor.

VALIDATION:

One validation study documented by author of the model.

LIMITATIONS:

- The model is applicable only to urban areas.
- There is no provision to include point sources.

MODEL:

Texas Episodic Model (TEM)

BASIC REFERENCE:

Porter, R.A. and Christiansen, J.H. "Two Efficient Gaussian Plume Models Developed at the Texas Air Control Board." Proceedings of the 7th NATO/CCMS International Technical Meeting on air Pollution Modeling, Airlie House, VA, September, 1976.

Christiansen, J.H., Users Guide to the Texas Episodic Model, Texas Air Control Board, May, 1976.

METHODOLOGY:

The Texas Episodic Model TEM is a short-term (10-minute to 24-hour averaging time) Gaussian plume model for prediction of concentrations of non-reactive pollutants from up to 300 elevanted point sources and up to 200 area sources. Concentrations are calculated for 1 to 24 scenarios of meteorological conditions, averaging time, and mixing height.

SOURCE GEOMETRY:

The area source is treated by the procedure described in Gifford and Hanna (1). The Plume rise for point source is according to Brigg's formula.

TRANSFER AND REMOVAL MECHANISMS:

Both physical and chemical removal processes are assumed to be exponential. The decay rate (half life) is specified by the user.

BASIC EQUATIONS:

The area source algorithm is due to Gifford and Hanna. (1) Each area source square is affected by its own diffuse emissions and those in the N area source squares directly upwind of it:

$$\chi = \left(\frac{2}{\pi}\right)^{1/2} \frac{(\Delta x/2)^{1-\cancel{b}\,d}}{U_0\, \underset{c}{\cancel{a}}\,(1-\cancel{b})} \left\{ Q_0 + \sum_{i=1}^{N} Q_i \left[(2i+1)^{1-\cancel{b}} - (2i-1)^{1-\cancel{b}} \right] \right\}$$

with ΔX = length of side of area source grid squares (m)
U_0 = surface wind speed (m/s)
Q_0 = area emission rate of square containing the receptor (gm/m^2/s)
Q_i = area emission rates of the upwind area sources (gm/m^2/sec)
a, b = stability and downwind distance-dependent parameters from the equation $\sigma_z = \underset{c}{\cancel{a}}\,\chi^{\cancel{b}\,d}$

The TEM employs steady-state bivariate Gaussian Plume point source logic. The concentration due to an elevated point source is given by:

$$\chi = \frac{10^6 Q}{\pi \sigma_y \sigma_z U} \exp\left(-y^2/2\sigma_y^2\right) \exp\left(-H^2/2\sigma_z^2\right) \exp\left(-\frac{0.692x}{UT_{1/2}}\right)$$

where χ = concentration (μg/m^3)
Q = emission rate (g/s)
U = wind speed at physical source height (m/s)
H = effective source height (m)
x = downwind distance (m)
y = crosswind distance (m)
$T_{1/2}$ = first-order pollutant decay half-life (s)
σ_y, σ_z = the standard deviations of the plume concentration distribution:

$$\sigma_y = \overset{a}{\cancel{c}}\, x^{\cancel{d}\,b} \qquad \sigma_z = \underset{c}{\cancel{a}}\, x^{\cancel{b}\,d}$$

with stability and downwind distance-dependent coefficients a, b, c, and d from Busse and Simmerman (2) and Turner (3). The wind speed U is the surface wind speed adjusted to the physical source height. Let K_y and K_z be defined by:

$$K_y = \frac{1000}{\sigma_y} \exp\left(-y^2/2\sigma_y^2\right)$$

$$K_z = \frac{1000}{\pi \sigma_z} \exp\left(-H^2/2\sigma_z^2\right)$$

Then, $$\chi = \frac{K_y K_z Q}{U}$$

K_y was calculated for each of the 1120 combinations of twenty downwind distances from 2 to 60 km, eight crosswind angles ($\tan^{-1} y/x$) from 0° to 7°, with δ varying from 1° to 5° depending on stability, and seven stability classes. For total vertical mixing below a mixing height of L meters:

$$\chi = \frac{10^6 Q}{\sqrt{2\pi}\,\sigma_y L U} \exp\left(-y^2/2\sigma_y^2\right)$$

This can be represented in the equation:

$$\chi = \frac{K_y K_z Q}{U} \quad \text{(decay term)}$$

by setting $K_z = 398L$

OUTPUT:

- Concentration for each receptor for various averaging times of up to 24 hours.
- List of the high five contributions to the concentrations at each receptor.

VALIDATION:

Limited experience with validation or comparison with observed data.

LIMITATIONS:

- Applicable to uncomplicated terrain.
- Reliability of the model is not established for cases where area sources dominate.
- Long-range transport is not considered.

REFERENCES:

(1) Gifford, F.A. and hanna S.R. "Modeling Urban Air Pollution," Atmos. Environ., 1973.

(2) Busse, A.D. and J.R. Zimmerman, "User's Guide for the Climatological Dispersion Model." Publication No. EPA-RA-73-024 (NTIS PB 227346/AS), Environmental Protection Agency, Research Triangle Park, NC 27711, Decemger, 1973.

(3) Turner, B.D. and Novak, J.H. "User's Guide for RAM," EPA-600/8-78-016, Environmental Protection Agency, Research Triangle Park, NC, November 1978.

MODEL:

Single Source Model (CRSTER)

BASIC REFERENCES:

Environmental Protection Agency. "User's Manual for Single Source (CRSTER) Model." Publication No. EPA-450/2-77-013 (NTIS PB 271360). Office of Air Quality Planning and Standards, Research Triangle Park, NC 27711, July, 1977.

METHODOLOGY:

CRSTER is a steady-state Gaussian Plume technique applicable to both rural and urban areas in uneven terrain. The purpose of the technique is: (1) to determine the maximum concentrations, for certain averaging times between 1 hour and 24 hours, over a one-year period from a single point source of up to 19 stacks, (2) to determine the meteorological conditions which cause the maximum concentrations, and (3) to store concentration information useful in calculating frequency distributions for various averaging times. The concentration for each hour of the year is calculated and midnight-to-midnight averages are determined for each 24-hour period.

SOURCE GEOMETRY:

Only point sources are considered. The plume rise is given by the Briggs formula. If the plume height exceeds the mixing height, the further downwind concentrations are assumed to be zero.

TRANSFER AND REMOVAL MECHANISMS: None

BASIC EQUATIONS:

$$\chi = \frac{Q}{2\pi u \sigma_y \sigma_z} g_1 g_3 \qquad \text{for } \sigma_z \leq 1.6L$$

$$\chi = \frac{Q}{\sqrt{2\pi}\, u L \sigma_y} g_1 \qquad \text{for } \sigma_z > 1.6L$$

χ = 0 (stability class 7)
L = mixing height (m)
H = (stack height + plume rise) - (difference in elevation between receptor and base of stack)

$$g_1 = \exp\left[-\tfrac{1}{2}\left(y/\sigma_y\right)^2\right]$$

$$g_3 = \sum_{n=-\infty}^{\infty}\left\{\exp\left[-\frac{1}{2}\left(\frac{2nL-H}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{2nL+H}{\sigma_z}\right)^2\right]\right\}$$

OUTPUT:

- Highest and second highest concentrations for the year at each receptor for averaging time of 1, 3, and 24 hours and a user selected averaging time of 2,4,6,8, or 12 hours.
- Annual arithmetic average at each receptor.
- Daily highest 1-hour and 24-hour concentrations.

VALIDATION:

Comparisons with observations at at least five separate power plants have been made.

LIMITATIONS:

- Not applicable to area or line sources.
- Not applicable to long-range transport of pollutants.

MODEL:

Climatological Dispersion Model (CDM) and CDMQC.

BASIC REFERENCE:

Busse, A.D. and J.R. Zimmerman. "User's Guide for the Climatological Dispersion Model." Publication No. EPA-RA-73-024 (NTIS PB 227346/AS), Environmental Protection Agency, Research Triangle Park, NC 27711, December 1973.

Brubaker, K.L., P. Brown, and R.R. Circillo. "Addendum to User's Guide for Climatological Dispersion Model." Publication No. EPA-450/3-77-015, Environmental Protection Agency, Research Triangle Park, NC 27711, May 1977.

METHODOLOGY:

The climatological dispersion model (CDM) determines long-term (seasonal or annual) quasi-stable pollutant concentrations at any ground level receptor using average emission rates from point and area source and a joint frequency distribution of wind direction, speed, and stability for the same period. The expanded version (CDMQC) include a statistical model based on Larson (1) to transform the average concentration data into expected geometric mean and maximum concentration values for several different averaging times.

SOURCE GEOMETRY:

Both point source and area source are treated in the model.

TRANSFER AND REMOVAL MECHANISMS:

Both physical and chemical removal mechanisms are assumed to be exponential and are treated by a user input half life.

BASIC EQUATIONS:

$$\chi_{\text{point}} = \frac{16}{2\pi} \sum_{n=1}^{N} \sum_{\ell=1}^{6} \sum_{m=1}^{6} Q_n \, \theta_{kn\ell m} \, S_{\ell m}(\rho_n) / \rho_n$$

$$\chi_{\text{area}} = \frac{16}{2\pi} \iint \left[\sum_{k=1}^{16} q_k(\rho) \sum_{\ell=1}^{6} \sum_{m=1}^{6} \phi_{k\ell m} \, S_{\ell m}(\rho) \right] d\rho$$

with

$$q_k(\rho) = \int_{\text{Sector } k} Q(\rho, \theta) \, d\theta$$

$$S_{lm}(\rho) = \frac{2}{\sqrt{2\pi}\,\sigma_z u_l} \exp\left[-\frac{1}{2}\left(\frac{H}{\sigma_z}\right)^2\right] \exp\left[-\frac{0.692\rho}{u_l T_{1/2}}\right]$$

for $\sigma_z \leq 0.8L$

$$S_{lm}(\rho) = \frac{1}{u_l L} \exp\left[-\frac{0.692\rho}{u_l T_{1/2}}\right]$$

for $\sigma_z > 0.8L$

$$\sigma_z = a\,\rho^b$$

a, b = functions of stability class (m) and downwind distance (ρ, three ranges of distance used: 100-500 m, 500-5000 m, and 5000-50,000 m

k_n = wind sector appropriate to the nth point source

Q_n = emission rate of the nth point source (g/s)

ρ_n = distance from the receptor to the nth point source (m)

q_K = emission rate of the area source per unit area and unit time (g/s-m^2)

ρ = distance from the receptor to an infinitesimal area source (m)

θ = angle relative to polar coordinates centered on the receptor

l = index identifying the wind speed class

m = index identifying the class of the Pasquil stability category

$\phi(k,l,m)$ = joint frequency function

z = height of receptor above ground level (m)

u_l = representative wind speed (m/s)

h = effective stack height of source distribution, i.e., the average height of area source emissions in the kth wind direction sector at radial distance from the receptor (m)

L = the afternoon mixing height (m)

$T_{1/2}$ = assumed half life of pollutant hours (s)

OUTPUT:

- Point area concentration rose for each receptor.
- One-month to one-year concentration using Larsen's procedure.
- Individual point, area source capability.

VALIDATION:

Calibration option is available in CDMQC and two validation studies are documented in user guide.

LIMITATIONS:

- Useable in urban areas only.
- Area source emission density must not vary rapidly from one area source to the next.

REFERENCES:

Larsen, R.I. "A Mathematical Model for Relating Air Quality Measurements to Air Quality standards." (NTIS PB 205277) Environmental Protection Agency, Research Triangle Park, NC 27711, November, 1971.

MODEL:

Gaussian-Plume Multiple - Source Air Quality Algorithm (RAM)

BASIC REFERENCE:

Turner, B.D. and Novak, J.H. "User's Guide for RAM," EPA-600/8-78-016, Environmental Protection Agency, Research Triangle Park, NC, November, 1978.

METHODOLOGY:

RAM is a short-term, gaussian, steady-state algorithm which estimates concentration of stable pollutants from urban point and area sources. Hourly meteorological data are used. Hourly concentration and averages over a number of hours can be estimated. Plume rise is calculated using Brigg's formula. Pasquill - Gifford dispersion equations and dispersion parameters are used. Concentrations from area sources are determined using the procedure described by Hanna and Gifford (1).

SOURCE GEOMETRY:

Both point and area sources are treated in this model. The area source is treated by the algorithm proposed by Gifford and Hanna (1).

TRANSFER AND REMOVAL MECHANISMS:

Both physical and chemical removal mechanisms are assumed to be exponential. The decay rate is specified by the user.

BASIC EQUATIONS:

The area source contributions are determined using the narrow-plume hypothesis similar to applications by Gifford and Hanna (1). Contribution from a single upwind area source

$$\chi_A = \frac{q}{u}\int_{x_1}^{x_2} f\,dx$$

integral evaluated numerically

X_1, X_2 = points of intersection of ray from receptor through area source in question
q = emission rate per unit area of the area source (g/s-m^2)
u = mean wind speed (m/s)

For stable conditions:

$$f = \frac{1}{\sqrt{2\pi}\,\sigma_z}\,g_2$$

$$x_{point} = \frac{Q}{2\pi u \sigma_y \sigma_z}\,g_1 g_2$$

For neutral or unstable conditions, $\sigma_z \leq 1.6L$

$$f = \frac{1}{\sqrt{2\pi}\,\sigma_z}\,g_3$$

$$x_{point} = \frac{Q}{2\pi u \sigma_y \sigma_z}\,g_1 g_3$$

For neutral or unstable conditions, $\sigma_z > 1.6 L$

$$f = \frac{1}{L}$$

$$\chi_{point} = \frac{Q}{\sqrt{2\pi}\, u L \sigma_y}\, g_1$$

In which

$$g_1 = \exp\left[-\frac{1}{2}\left(y/\sigma_y\right)^2\right]$$

$$g_2 = \exp\left[-\frac{1}{2}\left(\frac{z-H}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+H}{\sigma_z}\right)^2\right]$$

$$g_3 = \sum_{n=-\infty}^{\infty}\left\{\exp\left[-\frac{1}{2}\left(\frac{z-H+2nL}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+H+2nL}{\sigma_z}\right)^2\right]\right\}$$

OUTPUT:

- Hourly and average (24 hours) concentrations at each receptor.
- Limited individual source contribution list.
- Cumulative frequency distribution based on 24-hour averages and up to a year of data at a limited number of receptors.

VALIDATION:

A recent validation by Guildberg and Kern (2) and Kummler et al (3) has yielded the following results:

- The RAM (urban) model was found to be extremely inaccurate in predicting 24-hour SO_2 concentrations in urban environments.

- RAM (urban) model was found to be at least accurate at short distances down wind of elevated point sources.

- Cumulative frequency distribution predicted by RAM was better than SAI model.

LIMITATIONS:

- Applicable to flat or gently rolling terrain.
- Not applicable at short distances from the source.

REFERENCES:

(1) Gifford, F.A. and Hanna S. "Modeling Urban Air Pollution," Atmos. Environ., 1973.

(2) Guldberg, P.H. and Kern, C.W. "A Comparative Validation of RAM and PTMTP models for short-term SO_2 concentrations in two urban areas," J. Air. Pollut. Control Assoc., 28(9):907-911, September 1978.

(3) Kummler, R.H. Cho B., Roginski, G., Sinha, R., and Greenberg A. "A Comparative Validation of RAM and Modified SAI Models for Short-term SO_2 Concentrations in Detroit." J. Air Pollut. Control Assoc. 29(7): 720-723, July 1979.

MODEL:

Air Quality Display Model (AQDM)

BASIC REFERENCE:

"Air Quality Display Model," by TRW systems Group, Prepared for National Air Pollution Control Administration, DHEW, U.S. Public Health Service, Washington, D.C., November, 1969.

METHODOLOGY:

This model uses the Gaussian dispersion model which estimates annual arithmetic average of SO_2 and particulate concentrations at a fixed number of ground level receptors in an urban area. A statistical model based on Larsen (1) is used to transform the average concentration data from a limited number of receptors to expected geometric mean and maximum concentration values for several different averaging times.

SOURCE GEOMETRY:

Both point sources and area sources are treated in this model. Area source is treated by effective point source approximation. Each point source is characterized by its location (including height) and area source is characterized by the location and the area of the source: a single emission rate is specified for each source.

TRANSFER AND REMOVAL MECHANISMS:

The model treats one or two inert pollutants released simultaneously into the atmosphere. There is no provision for a chemically reacting pollutant. No provision for physical removal mechanisms *such as deposition) are present.

BASIC EQUATIONS:

$$\chi = \sum_{k=1}^{16} \sum_{l=1}^{6} \sum_{m=1}^{6} \phi_{klm} \chi_{klm}$$

where

$$\chi_{klm} = \frac{16}{2\pi x} \frac{2Q}{\sqrt{2\pi}\,\sigma_z u_l} \left(\frac{c-y}{c}\right) \exp\left[-\frac{1}{2}\left(\frac{H}{\sigma_z}\right)^2\right] \quad \text{for } x \leq x_L$$

$$= \frac{16}{2\pi x} \frac{Q}{u_l L} \left(\frac{c-y}{c}\right) \qquad \text{for } x > 2x_L$$

Linear Interpolation for $x_L < x < 2x_L$

x_L = 0.47L
y = cross wind distance between receptor sector centerline
c = sector width at receptor location

$\sigma_z(x) = ax^b + c$ a,b,c, functions of stability class

Q = emission rate (g/s)
H = plume height (m)
u = wind velocity (m/s)
ϕ_{klm} = relative frequency of occurence from stability wind rose.

To determine the wind field, 16 wind directions and 6 wind speed classes are used. The horizontal dispersion is calculated assuming a plume of 22.5° width.

OUTPUT:

The output of the models are concentrations at 225 guided receptor locations. One-month to one-year averaging time is simulated. Arbitrary averaging time 1-24 hours is done by Larsen's (1) procedure.

VALIDATION:

A large number of validations have been carried out. This model with Brigg's formula for plume rise typically estimates concentrations ranging from nearly correct to a factor or two high.

LIMITATIONS:

- Useable for urban area only.
- CDM/CDMQC generally prefered over AQDM.

REFERENCES:

(1) Larsen, R.I. "A Mathematical Model for Relating Air Quality Measurements to Air Quality Standards." (2) Office of Air Programs Publication #AP-89 (NTIS PB 205277), Environmental Protection Agency, Research Triangle Park, NC 27711, November 1971.

MODEL:

Texas Climatological Model (TCM)

BASIC REFERENCE:

Porter, R.A. and Christiansen, J.H. "Two Efficient Gaussian Plume Models Developed at the Texas Air Control Board." Proceedings of the 7th NATO/CCMS International Technical Meeting on Air Pollution Modeling, Airlie House, Va., September, 1976.

Christiansen, J.H. and Porter, R.A., Users Guide to the Texas Climatological Model, Texas Air Control Board, Austin Texas, May 1976.

METHODOLOGY:

The Texas climatological model predicts long-term arithmetic mean concentrations of non-reactive pollutants from point and area sources. The area sources are handled by the algorithm proposed by Gifford and Hanna (1). The model uses the steady-state Gaussian plume point source logic to determine the concentration distribution.

SOURCE GEOMETRY:

Both point and area sources are treated in this model. The area sources are treated by the algorithm proposed by Gifford and Hanna (1).

TRANSFER AND REMOVAL MECHANISMS:

Both physical and chemical removal mechanisms are assumed to be exponential. The decay rate is according to the user defined input half-life.

BASIC EQUATIONS:

The concentration from a point source is given by the following equation:

$$\chi = Q_i \sum_{m=1}^{6} \left[K(x,H,m)\, \phi(k,m)/U_m \right] \cdot \text{(decay term)}$$

where

$$K(x,H,m) = 32 \times 10^6 \Big/ \left[(2\pi)^{3/2} \times \sigma_z \right] \exp\left(-H_i^2/2\sigma_z^2\right)$$

(K is precalculated for 20 distances, 9 effective source heights, and six stability classes.)

U_m is a wind speed characteristic of an entire stability class, and is computed in the model by the equation:

$$U_m = \left[\sum_{k=1}^{16} \sum_{l=1}^{6} \phi(k,l,m) \right] \times \left[\sum_{k=1}^{16} \sum_{l=1}^{6} \phi(k,l,m)/u_l \right]^{-1}$$

with χ = concentration, $\mu g/m^3$
k = the wind sector index appropriate to source i at the receptor
l = wind speed class index
m = stability class index
ϕ = meteorological joint frequency function
Q_i = emission rate of source i, g/s
H_i = effective height of source i, m
σ_z = standard deviation of vertical Gaussian concentration distribution, m

$T_{1/2}$ = half-life for first-order pollutant decay s
U_ℓ = central wind speed of class l, m/s
X = downwind distance, m

The TCM uses steady-state Gaussian plume point source logic, with the crosswind distribution averaged across 22.5° azimuthal sectors. The only meteorological input required for area source calculations is the mean wind speed, but the point source calculations require a meteorological joint frequency function with sixteen 22.5° wind sectors, sic wind speed classes (0-3, 4-6, 7-10, 11-16, 17-21, and 21 knots), and six stability classes (Turner classes A, B, C, D (day), D (night), and E plus F).

The concentration due to an area source is given by:

$$\chi_A = \frac{FQ}{u}$$

where χ_A = concentration $\mu g/m^3$
Q = area source emission rate in the vicinity of the receptor ($\mu g/m^2\text{-}s$).
U = mean ground level wind speed (m/s)
F = dimensionless constant

Gifford and Hanna have suggested a value of F = 50 for SO_2 and 225 for total suspended particulate (TSP). The area emission rate, Q, is determined by averaging the emissions in the area source square containing the receptor and in the neighboring squares. The extent of the region around each receptor to use for emission rate averaging is an input parameter.

OUTPUT:

Arithmetic mean concentration for the averaging time of the climatological input and emission data (1 month to 1 year). In addition, the following outputs are available:

- Listing of concentration for an arbitrarily spaced square grid of up to 50 x 50 elements.
- Plot of grid concentrations.
- Listing of the five high contributors to the concentration (by % concentration) at each grid point.

VALIDATION:

The model is self-calibrating with input of field receptor observations. Poor correlations have been shown where area sources dominate.

LIMITATIONS:

- Applicable for flat terrains, relatively constant emissions.
- Poor correlation where area sources dominate.

REFERENCES:

Gifford, F.A. and Hanna S.R. "Modeling Urban Air Pollution," Atmos. Environ. 1973.

MODEL:

Valley

BASIC REFERENCE:

Edward W. Burt, "Valley Model User's Guide," U.S. Environmental Protection Agency, Research Triangle Park, NC, EPA-450/2-77-018, 1977 (NTIS accession number PB 274-054).

METHODOLOGY:

This algorithm is a steady-state, univariate Gaussian plume dispersion algorithm designed for estimating either 24-hour or annual concentrations resulting from emissions from up to 50 (total) point and area sources. Calculations of ground-level pollutant concentrations are made for each frequency designed in an array defined by six stabilities, 16 wind directions, and six wind speeds for 112 program-designed receptor sites on a radial grid of variable scale. Empirical dispersion coefficients are used and include adjustments for plume rise and limited mixing. Plume heights are adjusted according to terrain elevations and stability classes.

SOURCE GEOMETRY:

Both point sources and area sources (up to 50) can be considered. The emission characteristics of the source are input as average values for each source for the selected averaging time period.

TRANSFER AND REMOVAL MECHANISMS:

The model, at the user's option, considers a first order removal process.

BASIC EQUATIONS: Not available.

OUTPUT:

The program sets up a grid of 112 receptors and the 24-hour concentrations at the recetpors are calcualted using the long-term stability wind data (STAR).

VALIDATION:

A validation study indicated the calculated concentration to be within a factor of 2 of the second highest observed concentration (for four out of six monitored sites).

MODEL:

Point source models: PTMAX, PTDIS and PTMTP

BASIC REFERENCES:

D.B. Turner and A.D. Busse, "User's Guide to the Interactive Versions of Three Point Source Dispersion Programs: PTMAX, PTDIS, and PTMTP." Preliminary Draft, Meteorology Laboratory, Environmental Protection Agency, Research Triangle Park, NC 27711, 1973.

METHODOLOGY:

These three models use Brigg's formula for plume rise and Pasquill-Gifford dispersion methods to estimate hourly mean concentrations from stable pollutants.

PTMAX: Performs an analysis of the maximum short-term concentrations from a single point source as a function of stability and wind speed. The final plume height is used for each computation.

PTDIS: Estimates short-term concentrations directly downwind of a point source at distances specified by the user. The effect of limiting vertical dispersion by a mixing height can be included, and gradual plume rise to the point of final rise is also considered. An option allows the calculation of isopleth half-widths for specific concentrations at each downwind distance.

PTMTP: Estimates for a number of arbitrarily located receptor points at or above ground-level, the concentration from a number of point sources. Plume rise is determined for each source. Downwind and crosswind distances are determined for each source-receptor pair. Concentrations at a receptor from various sources are assumed additive. Hourly meteorological data are used; both hourly concentrations and averages over any averaging time from one to 24 hours can be obtained.

SOURCE GEOMETRY:

Only point sources are considered.

TRANSFER AND REMOVAL PROCESS:

Not considered.

BASIC EQUATIONS:

Not Available.

OUTPUT:

The models estimate the short-term concentrations at receptor locations. An option allows PTDIS to calcualte isopleth half widths of specific concentrations.

VALIDATION:

PTMTP has been validated by Guzewich and Pringle (1) and was found to be satisfactory in predicting the dispersion of gases from an elevated buoyant source. (89% of all predicted values were within a factor of 3 from measured concentrations.)

REFERENCES:

Guzewich, D.G., Pringle, W.J.B., "Validation of EPA-PTMTP short-term Gaussian Dispersion Model," J. Air Pollut. Control Assoc. 27(6):540-542, June 1977.

MODEL:

Point Area, Line Source Algorithm (PAL)

BASIC REFERENCE:

William B. Petersen, "User's Guide for PAL — A Gaussian-Plume Algorithm for Point, Area, and Line Sources." EPA-600/4-78-013 (NTIS PB 281-306), Environmental Protection Agency, Research Triangle Park, NC 27711, 1978.

METHODOLOGY:

Point, Area, Line source algorithm. This short-term Gaussian steady-state algorithm estimates concentrations of stable pollutants from point, area, and line sources. Computations from area sources include effects of the edge of the source. Line source computations can include effects from a variable emission rate along the source. The algorithm is not intended for application to entire urban areas, but for smaller scale analysis of such sources as shopping centers, airports, and single plants. Hourly concentrations are estimated and average concentrations from 1 hour to 24 hours can be obtained.

SOURCE GEOMETRY:

This model is designed to consider point, area, horizontal line, horizontal area, and specialized line and curve sources. Area sources are square or rectangular. Integration over an area source is done by considering finite line sources perpendicular to mean wind vector.

TRANSFER AND REMOVAL MECHANISMS:

No removal or transfer mechanisms are considered.

BASIC EQUATIONS:

Not available.

OUTPUT:

Short-term concentrations (up to 24 hours) at various receptor locations (30) can be obtained as an output.

VALIDATION:

Not available.

LIMITATIONS:

Because of the detail with which PAL heats the sources, application of PAL to larger periods of time (>24 hours) or long-range transport (>a few hundred meters) would be expensive.

MODEL:

SAI Air shed model.

BASIC REFERENCE:

J. Ames et. al., "The User's Manual for the SAI - Air Shed Model," EM 78-79, Environmental Protection Agency, Research Triangle Park, NC 27711, August 1978.

METHODOLOGY:

The SAI Air shed model is capable of calculating concentrations of both reactive and inert pollutants through a simulation of the numerous physical and chemical processes that take place in the atmosphere. The model has the following features:

- Transport of pollutants by advection (including wind shear).
- The diffusion of pollutants is treated using an eddy diffusivity coefficient.
- The surface removal processes are treated by considering the transport of pollutants to the surface and the uptake by the surface.
- A carbon-bond mechanism is included to take into account the reactivity of a pollutant.

SOURCE GEOMETRY:

Point source emissions are considered. The effective plume height is calculated using Brigg's formula. Other emissions include non point ground level emissions (stationary as well as mobile).

TRANSFER AND REMOVAL MECHANISMS:

Both physical and chemical transfer processes are considered.

BASIC EQUATIONS:

Not available.

OUTPUT:

SAI model provides the following output:

- Vertical concentration profile.
- Ambient concentratoin maps.

VALIDATION:

The SAI model has been applied at several locations and has yeilded very formidable results. A recent comparative validation study by Kummler et. al. (1) claims SAI model to be a better simulator than RAM.

LIMITATIONS:

For long-range pollutant transport, several grids may be needed, thus increasing the computing time.

REFERENCES:

Kummler, R.H., Cho B., Roginski, G., Sinha R., and Greenberg A., "A Comparative Validation of RAM and Modified SAI Models for Short-term SO_2 Concentrations in Detroit," J. Air Pollut. Control Assoc. 29(7):720-723, July 1979.

MODEL:

Atmospheric Transport Model (ATM).

BASIC REFERENCE:

Culkowski, W.M. & M.R. Patterson. 1976. A Comprehensive Atmospheric Transport and Diffusion Model. Oak Ridge National Laboratory, ORNL/NSF/EATC-17.

METHODOLOGY:

ATM is a long term average (monthly, seasonal, or annual) Gaussian plume model for prediction of concentrations of non-reactive pollutants. ATM uses the joint frequency distribution of wind direction, speed and stability over the averaging time, similar to the approach of CDM. Wet and dry deposition processes are explictly simulated. The model is designed for incorporation in UTM - The Unified Transport Model.

SOURCE GEOMETRY:

Point, area and line sources may be simulated in ATM. Area and line sources are decomposed into a number of point sources.

TRANSFER AND REMOVAL MECHANISMS:

Chemical reaction processes are not simulated by ATM. Wet and dry deposition processes are treated via the source depletion approach, gravitational settling via the titled plume approach. The settling velocity, deposition velocity and washout coefficient are the required input parameters affecting deposition.

BASIC EQUATIONS:

$$X_i = \sum_{p=1}^{N_s} \sum_{r=1}^{N_w} \sum_{j=1}^{N_j} \frac{2.032\, F_{pr}(\theta_j)\, Q_{ipr}(x_j)}{\sigma_p(x_j)\, u_r\, x_j} \exp\left[\frac{-h_{pr}^2}{2\sigma_p^2(x_j)} \right]$$

X_i = ground level air concentration of pollutant i, in direction , at a distance of x from the source,

θ = one of sixteen cardinal wind direction sectors (N, NNE, NE, etc.)

r = one of eight wind speed classes,

$F_{pr}(\theta)$ = fraction of time during which the wind blows from direction sector , with wind speed class r, and stability class s (Stability Wind Rose),

Q_{ipr} = point source strength for pollutant i, modified by depletion due to fallout and washout occurring at distances less than x,

N_s = number of stability classes,

N_w = number of wind speed classes,

σ_p = vertical dispersion parameter appropriate for stability class p and distance x,

h_{pr} = effective stack height,

p = one of six wind stability classes (A, B, C, D, E, F) changing from extremely unstable to moderately stable.

u_r =

$$Q = Q_0 \left\{ \exp\left[-\sqrt{\frac{2}{\pi}}\, \frac{v_g}{u} \int_0^x \frac{e^{-h^2/2\sigma_z^2}}{\sigma_z}\, dx' - \frac{\lambda x}{u} \right] \right\}$$

v_g = deposition velocity

λ = washout coefficient

OUTPUT:

Concentration increments from each point and area source, deposition rate increments from each point and area source, total wet and dry deposition and concentration, monthly average.

VALIDATION:

SO_2 concentrations resulting from two coal-fired power plants compared with model results satisfactorily ($r = 0.80$).

LIMITATIONS:

- Shares the limitations of all Gaussian plume models
- source depletion approach is inexact and fails to account for the fact that pollutants are lost at the ground surface
- No provision for chemical reactions causing plume depletion
- Long-term averaging inhibits interface with runoff models (NPS, WHTM) which respond hourly

REFERENCES

1. Eschenroeder, A.Q., Models for Predicting Air Quality, Chapter 5, In Ozone and Other Photochemical Oxidants. Washington, DC: National Academy of Sciences, 1977.

2. Barth, D.S., Federal Motor Vehicle Emission Goals for CO, HC, and NOH$_x$ Based on Desired Air Quality Levels. J. Air Pollut. Control Assoc. 20:519-523, 1970.

3. Bruntz, S.M., W.S. Cleveland, B. Kleiner, and J.L. Warner. The Dependence of Ambient Ozone on Solar Radiation, Wind, Temperature, and Mixing Height. Pages 125-128. In preprints. Symposium on Atmospheric Diffusion and Air Pollution of the American Meteorological Society, Santa Barbara, September 9-13, 1974; Boston: American Meteorological Society, 1974.

4. Chang, T.Y., and B. Weinstock, Rollback Modeling for Urban Air Pollution Control, pp. 184-189. In preprints. Symposium on Atmospheric Diffusion and Air Pollution of the American Meteorological Society, Santa Barbara, California, September 9-13, 1974, Boston: American Meteorological Society, 1974.

5. Chang, T.Y., and B. Weinstock, Urban CO Concentrations and Vehicle Emissions. J. Air Pollut. Control Assoc. 23:691-696, 1973.

6. Hamming, W.J., R.L. Chass, J.E. Dickinson, and W.G. MacBeth, Motor Vehicle Control and Air Quality: The Path to Clean Air for Los Angeles, Paper 73-73 presented at the 66th Annual Meeting of the Air Pollution Control Association, Chicago, IL, June 24-28, 1973. 19 pp.

7. McCollister, G.M., and K.R. Wilson, Linear Stochastic Models for Forecasting Daily Maxima and Hourly Concentrations of Air Pollutants. Atmos. Environ. 9:417-423, 1975.

8. Mertz, P.H., L.J. Pinter, and P.R. Ryason, Aerometric Data Analysis--Time Series Analysis and Forecast and an Atmospheric Smog Diagram. Atmos. Environ. 6:319-342, 1972.

9. Peterson, J.T., The Calculation of Sulfur Dioxide Concentrations Over a Metropolitan Area by Using Empirical Orthogonal Functions. American Institute of Aeronautics and Astronautics, Paper 70 113, January 1970, 7 pp.

10. Schuck, E.A., A.P. Altshuller, D.S. Barth, and G.B. Morgan, Relationship of Hydrocarbons to Oxidants in Ambient Atmospheres. J. Air Pollut. Control Assoc. 20:297-302, 1970.

11. Schuck, E.A., and R.A., Papetti. Examination of the Photochemical Air Pollution Problem in the Southern California Area, Appendix D. In Technical Support Document for the Metropolitan Los Angeles Intrastate Air Quality Control Region Transportation Control Plan Final Promulgation. San Francisco: U.S. Environmental Protection Agency, Region IX, October 30, 1973. 19 pp.

Reproduced from best available copy.

12. Trijonis, J.C., Economic Air Pollution Control Model for Los Angeles County in 1975. General least cost-air quality model. Environ. Sci. Technol. 8:811-826, 1974.

13. Gifford, F.A., and S.R. Hanna, Modelling Urban Air Pollution. Atmos. Environ. 7:131-136, 1973.

14. Hanna, S. R., A Simple Dispersion Model for the Analysis of Chemically Reactive Pollutants. Atmos. Environ. 7:803-817, 1973.

15. Hanna, S.R., A Simple Method of Calculating Dispersion from Urban Area Sources. J. Air Pollut. Control Assoc. 21:774-777, 1971.

16. Hanna, S.R., Author's reply. [To discussion of "A simple dispersion model for the analysis of chemically reactive pollutants."] Atmos. Environ. 8:529-530, 1974.

17. Lamb, R.G., and J.H. Seinfeld, A simple dispersion Model for the Analysis of Chemically Reactive Pollutants. Atmos. Environ. 8:527-529, 1974 (discussion).

18. Egan, B.A. and J.R. Mahoney, Applications of a Numerical Air Pollution Transport Model to Dispersion in the Atmospheric Boundary Layer. J. Apple. Meteorol. 11:1023-1039, 1972.

19. Egan, B.A. and J.R. Mahoney, Numerical Modeling of Advection and Diffusion of Urban Area Source Pollutants. J. Apple. Meteorol. 11:312-322, 1972.

20. Eschenroeder, A.Q. and J.R. Martinez, Concept and Applications of Photochemical Smog Models. Adv. Chem. Ser. 113:101-168, 1972.

21. Eschenroeder, A.Q. and J.R. Martinez. Mathematical Modeling of Photochemical Smog. GRC Report IMR-1210. Santa Barbara California: General Research Corporation, 1969, 11 pp.

22. Eschenroeder, A.G., J.R. Martinez, and R. Nordsieck, A View of Future Problems in Air Pollution Modeling, pp. 1013-1027. In Proceedings of the 1972 Summer Computer Similuation Conference. Montvale, NJ: American Federation of Information Processing Societies, 1972.

23. Friedlander, S.K. and J.H. Seinfeld, A Dynamic Model of Photochemical Smog. Environ. Sci. Technol. 3:1175-1182, 1969.

24. Karplus, W.J., G.A. Bekey, and P.J. Pekrul, Atmospheric Diffusion of Air Pollutants: Analog Computer Study. Ind. Eng. Chem. 50:1657-1660, 1958.

25. Lamb, R.G. and M. Neiburger. An Interim Version of a Generalized Urban Ait Pollution Model. Atmos. Environ. 5:239-264, 1971.

26. Mahoney, J.R. and B.A. Egan, A Mesoscale Numerical Model of Atmospheric Transport Phenomena in Urban Areas, pp. 1152-1157. In H.M. Englund and W.T. Beery, Eds. Proceedings of the Second International Clean Air Congress. New York: Academic Press, 1971.

27. Martinez, J.R., R.A. Nordsieck, and A.Q. Eschenroeder, Morning Vehicle-start Effects on Photochemical Smog. Environ. Sci. Technol. 7:917-923, 1973.

28. Ulbrich, F.A., A Predictive Air Pollution Control for the Los Angeles Basin. Socio-Econ. Plan. Sci. 1:423-440, 1968.

29. Wayne, L.R., M. Danchick, M. Weisburd, A. Kokin, and A. Stein, Modeling Photochemical Smog on a Computer for Decision-Making. J. Air Pollut. Control Assoc. 21:334-340, 1971.

30. Sutton, O., 1953: Micrometeorology, McGraw Hill, New York.

31. ASME, 1968: Recommended Guide for the Prediction of the Dispersion of Airborne Effluents, M. Smith (ed.), New York, NY.

32. Gifford, F.A., Atmospheric Transport and Dispersion Over Cities. Nuclear Safety, 13:391-402, 1978.

33. Briggs, G.S., Some Recent Analyses of Plume Rise Observation. Second International Clean Air Congress, International Union of Air Pollution Prevention Associatins, Washington, DC, 1970.

34. Pasquill, F., The Estimation of the Dispersion of Windborn Material. Meteorology Magazine, 90:1063,33-49, 1961.

35. Turner, D., Workbook of Atmospheric Dispersion Estimates. US Department of Health, Education, and Welfare, Public Health Service Publication No. 999-AP-26, 1967.

36. Bigg, E., C. Ayers, and D. Turvey, Measurement of the Dispersion of a Smoke Plume at Large Distances from the Soruce. Atmos. Environ. 12:1815-1818, 1978.

37. Lucas, D., The Atmospheric Pollution of Cities, International Journal of Air Pollution, 1:71-86, 1958.

38. Aravamudan K., M. Bonazountas, A. Eschenroeder, D. Gilbert, L. Nelken, K. Scow, R. Thomas, W. Tucker, and C. Unger, An Environmental Partitioning Model for Risk Assessment of Chemicals: Part 1. Narrtive Descriptions of the Methodology, Arthur D. Little, Inc. Report to US Environmental Protection Agency, Office of Toxic Substances, Contract 68-01-3857 in review, 1979.

39. Ermak, D., An Analytical Model for Air Pollutant Transport and Deposition from a point source, Atmos. Environ. 11:231-237, 1977.

40. Pasquill, F., Atmospheric Diffusion. Second Edition, John Wyley and Sons, New York, 259-274, 1974.

41. Slade, D., Meteorology and Atomic Energy, US Atomic Energy Commission Office of Information Services, p. 73, 1968.

42. Neeley, W., Predicting the Flux of Organics Across the Air/Water Interface, in Control of Hazardous Spills: Proceedings of 1976 National Conference on Control of Hazardous Material Spills, US Environmental Protection Agency and Oil Spill Control Association of America, 197-200, 1976.

43. Briggs, G., Chimney Plumes in Neutral and Stable Surroundings. Atmos. Environ., 6:507-510, 1972.

APPENDIX B

SIZE DISTRIBUTION OF VARIOUS AEROSOLS

The surface area distribution of ambient rural and urban aerosols, fly ash particulate from electric utility boilers using electrostatic precipitation to control emission, and fly ash particulate from uncontrolled electric utility boilers is presented in Figures B-1, B-2 and B-3.

The distribution curves were calculated from either particle mass or number distributions presented in the literature. The number distribution was converted to a surface area distribution by numerical integration in .25 steps of log radius and assuming spherical symmetry as follows:

$$\sum_{r=1}^{r=n} \Delta S_{r_{n-1}}^{r_n} = \frac{dN}{\log r_n/r_{n-1}} \times \log r_n/r_{n-1} \times 4\pi \left(\frac{r_n + r_{n-1}}{2}\right)^2$$

where ΔS = surface area of particles contained in the interval $r_n - r_{n-1}$

N = number of particles contained in the interval $r_n - r_{n-1}$

r = radius of particle

In the case of fly ash particulate mass distribution the number of particles was first determined by numerical integration in steps of .25 log units of particle diameter assuming a density[2] (D) of 2.6 g/cm^3 as follows:

$$\Delta N_{d_1}^{d_2} = \left[\Delta M_{d_1}^{d_2} \Big/ D\right] \div \left[\frac{4}{3}\pi \left(\frac{r_2 = r_1}{2}\right)^3\right]$$

Surface Area Distribution vs. Particle Size

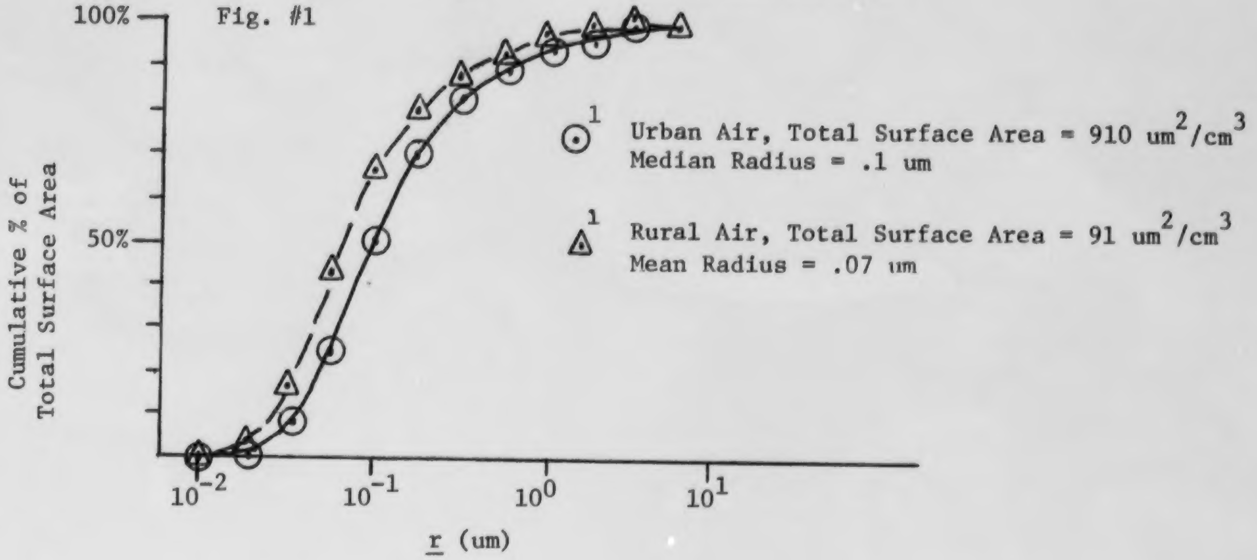


Fig. #1

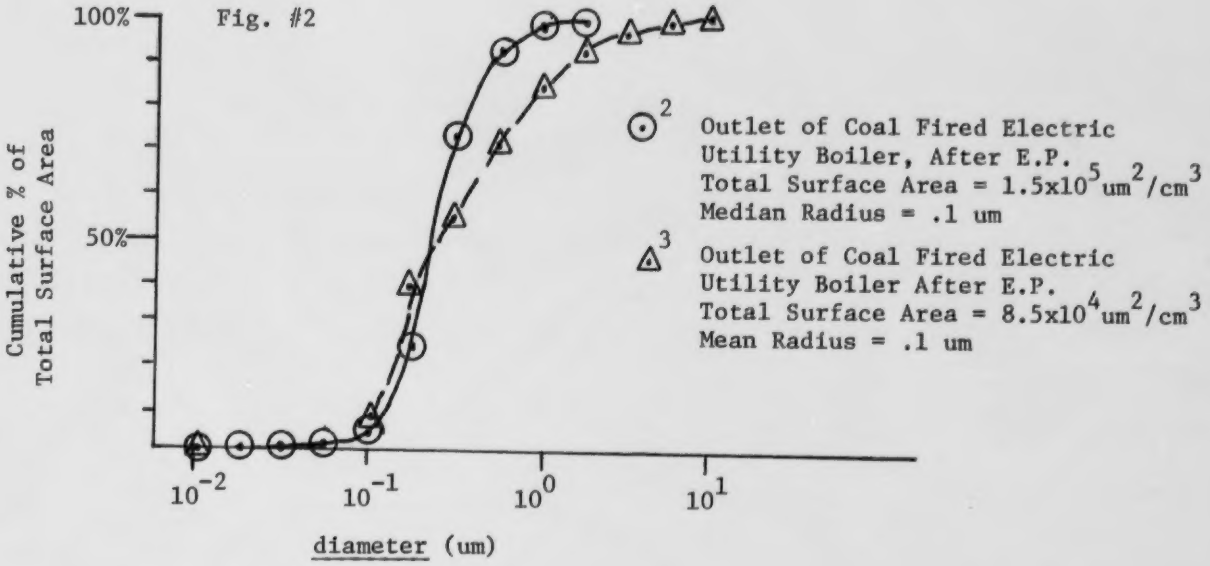


Fig. #2

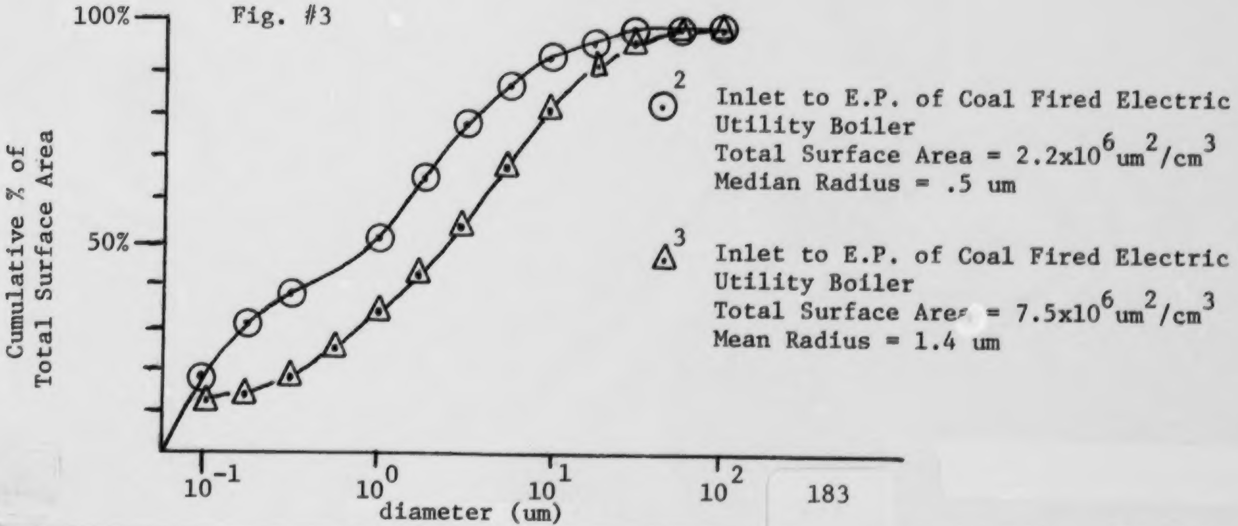


Fig. #3

where N = number of particles

M = mass per

d = particle diameter

D = particle density

Subsequent calculation of the surface area is analogous to the number distribution conversion.

The ambient rural and urban aerosol values have a $d_A{}^{50}$ radius of .07 to .1 um. The total surface area found in the former calculations is in agreement with values presented by Junge[4] for rural and urban areas. The fly ash electrostatic precipitation controlled emissions also have a $d_A{}^{50}$ radius value of .1 um. Values calculated from two independent sources are in good agreement. The uncontrolled fly ash emissions were found to have $d_A{}^{50}$ radius value of .5 and 1.4 um. The two values are calculated from two independent sources and present a useful $d_A{}^{50}$ radius range as the particulate size range is probably boiler dependent.

Farm Tillage Soil Emission Mean Surface Area Diameter

Calculation Method: Given the particle size distribution as a function of weight, the data is converted to a distribution as a function of surface area by numerical integration in .25 log unit steps. The calculation equation is:

$$S_i = \left[\frac{m_i/d}{\left[3\ x\left(\frac{r_n = r_{n+1}}{2}\right)\right]}\right] \div \sum_{n=1}^{n=16} \left[\frac{m_i/d}{\left[3\ x\left(\frac{r_n + r_{n+1}}{2}\right)\right]}\right] =$$

$$\frac{m_i/(r_n + r_{n+1})}{\sum_{n=1}^{n=16} m_i/(r_n + r_{n+1})}$$

where S_i = % of surface area in the interval r_n to r_{n+1}

m_i = % of mass in the interval r_n to r_{n+1}

d = density of soil, 2.5 g/cm^3

n to n+1 = .25 log unit steps of particle diameter beginning with .01 um

The surface area distribution curves for two cases presented by Cowherd[5] (1974) are plotted on Figure B-4. The $d_a{}^{50}$ radius is ~ .5 um.

REFERENCES

1. Junge, C.E., Air Chemistry and Radioactivity, pg. 117, Academic Press, N.Y. (1963).

2. Anath, K.P. et al., "Particle Emission Reactivity", EPA Report #EPA-600/2-76/257, pg. 34-38 (1976).

3. Nichols, G. et al., "Particulate Collection Efficiency Measurements on Three Electrostatic Precipitators," EPA Report #EPA-600/2-75-056, pg. 28-29 (1975).

4. Junge, C.E., "Basic Considerations About Trace Constituents in the Atmosphere as Related to the Fate of Global Pollutants," Fate of Pollutants in the Air and Water Environments, pg. 7-23, Wiley & Sons, N.Y. (1977).

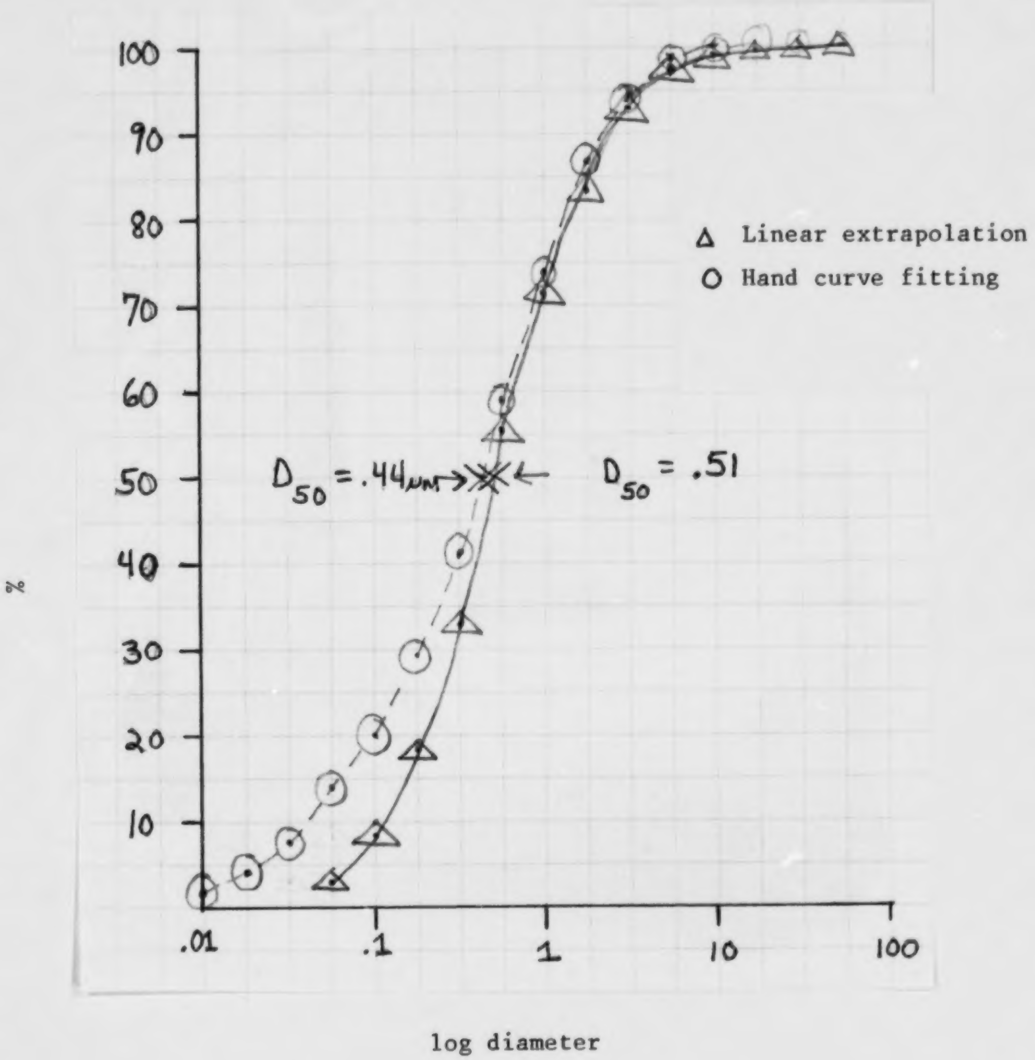
5. Cowherd, C., Jr., K. Axetell, Jr., C.M. Guenther, and G. Jutze. "Development of Emission Factors for Fugitive Dust Sources," Final Report, Midwest Research Institute for U.S. Environmental Protection Agency, Publication No. EPA-450/3-74-037, NTIS No. PB 238262/AS, June 1974.

FIGURE B-4

% SURFACE AREA OF PARTICLES

Farm Soil Emissions

Reproduced from best available copy.



APPENDIX C

ALWAS PROGRAM LISTING

DIDOT

```
      BLOCK DATA
      COMMON /ORIGIN/ X0,Y0,UNIT,NEAST,NNORTH,NPOLLS
      COMMON /RECEPS/ INDEX(225),INDPPT(20),KATCH(20),
     X                NEXAMS(10),KNTEX
      COMMON /PTSRCS/ QXY(4,10),PARAMS(5,10),HTABL(3,10,10),
     X                DISTBL(3,10,10),COORDS(2,10),IFLAG(10),KNTPTS
      COMMON /ARSRCS/ VGRID,ACROSS,UP,QXYS(5,50),KNTARS
      COMMON /CITIES/ QXYCIT(5,5),SIGZ0(6,5),XTENT(6,5),IFLAGC(5),KNTCTY
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)
      COMMON /IDENTS/ KEY1(6),DOLLAR
      COMMON /METDAT/ IDAYS(366),ANGL(24),U(24),KST(24),HL(24),TEMP(24),
     X                IRAIN(25,5),TSNOW
      COMMON /DISKOP/ DAYDEP(2,225),PPTCON(2,20,24),DPLCON(2,20,24)
      COMMON /ANSTAT/ ANNAVG(2,20,2),KDAYS
      COMMON /VALUES/ ADJWD(2),TAU(2),
     X                UDEPOS(2),USETTL(2),SCAVNG(2),BKGRND(2),RUFLEN
      COMMON /ARANGL/ EPS(24),IEPS(24),ISIGNX(24),ISIGNY(24),
     X                ITABX(24),ITABY(24),INCRX(24),INCRY(24)
      COMMON /ACONST/ EPSLIM(10),AAR(6),BAR(6),ITABLE(5,10)
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,
     X                T,P,YCOEFF(4),YPOWER(4),ZCOEFF(4),ZPOWER(4),
     X                R2PI,ROOT2,ROOTPI,
     X                J1STAB(7),J2STAB(7)
      COMMON /TAB001/ HLOW,HHIGH,HDEL,XLTAB(50,4),
     X                XLOW,XHIGH,XDEL,SYTAB(50,4),SZTAB(50,4),
     X                ELOW,EMAX,EDEL,ETAB(100),NCLASS,NHT,NXC,NEXP
      COMMON /CHITAB/ HRMAXP(2,5),HRMAXA(2,5),HRMAXC(2,5),
     X                MAXHRP(2,5,2),MAXHRA(2,5,2),MAXHRC(2,5,2),
     X                DAYMAX(2,10),MAXDAY(2,10)
      COMMON /MCONST/ KNTDAY(12),NDAY,IYEAR,IMONTH,IDAYNO,L(5)
      COMMON / TEST / IDBUG,LUNDBG
      REAL*8 KEY1
      LOGICAL IDBUG
      DATA IDBUG/.FALSE./,LUNDBG/4/
      DATA INDEX,NNORTH,NEAST,KATCH/247*0/,UNIT,X0,Y0/3*0.0/
      DATA INDPPT,IRAIN,NPPT,LUNPPT/151*0/,KNTEX,NEXAMS/11*0/
      DATA QXY,PARAMS,HTABL,DISTBL/690*0.0/
      DATA QXYS,VGRID,ACROSS,UP/253*0.0/,COORDS/20*0.0/
      DATA LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS/5,6,7,8,9/
      DATA DOLLAR/2H $/
      DATA KEY1/6HPTSRCE,6HARSRCE,6HCITY  ,6HCATCHS,6HMETCTL,6HCNTROL/
      DATA IDAYS,KNTPTS,KNTARS,KNTCTY/369*0/,ANGL,U,HL,TEMP,EPS/120*0.0/
      DATA DAYDEP,PPTCON,DPLCON/2370*0.0/
      DATA ANNAVG/80*0.0/,KDAYS/0/
      DATA L/ -50,-500,32767,50,5/
      DATA USETTL,UDEPOS,SCAVNG,BKGRND,RUFLEN/ 9*0.0/,
     X     TAU/2*9999.999/,TSNOW/32.0/
      DATA J1STAB/4*1,2,3,3/
      DATA J2STAB/1,1,2,3,4,4,4/
      DATA KST,IFLAG,IEPS,ISIGNX,ISIGNY,ITABX,ITABY,INCRX,INCRY/202*0/
      DATA ITABLE/0,0,0,0,0, 0,0,0,0,1, 0,0,0,1,1, 0,0,1,1,1, 0,0,1,1,2,
     1            0,1,1,2,2, 0,1,1,2,3, 0,1,2,2,3, 0,1,2,3,3, 0,1,2,3,4/
      DATA YCOEFF/0.40,0.36,0.32,0.31/,YPOWER/0.91,0.86,0.78,0.71/
      DATA ZCOEFF/0.40,0.33,0.22,0.06/,ZPOWER/0.91,0.86,0.78,0.71/
      DATA AAR/.4,.33,.22,.15,.06,.06/
      DATA BAR/.91,.86,.80,.75,.71,.71/
      DATA EPSLIM/.12427,.16511,.24504,.35884,.46373,
     1           .55868,.64350,.69482,.71890,.78540/                    BLK00580
      DATA WIDTH/2*0.872665,0.6981317,0.523599,3*0.3406585/              BLK00590
      DATA WSPDS/1.03,1.544,2.57,3.09,4.12,5.15,6.18,7.21,7.72,8.24/    BLK00600
      DATA KNTDAY/0,31,59,90,120,151,181,212,243,273,304,334/           BLK00610
      DATA NDAY,IYEAR,IMONTH,IDAYNO/4*0/                                 BLK00620
      DATA NHT/50/,HLOW/100.0/,HDEL/62.5/                                BLK00630
      DATA NXC/50/,XLOW/2000.0/,XDEL/200.0/                              BLK00640
      DATA NEXP/100/,NCLASS/4/,ELOW/0.0/,EDEL/.075/                      BLK00650
      DATA G/9.80616/, PION4/0.785398/,T/283.0/,P/960.0/                 BLK00660
      DATA R2PI/2.506628/, ROOTPI/1.772454/,ROOT2/1.414214/              BLK00670
      DATA QXYCIT,SIGZO,XTENT/85*0.0/,IFLAGC/5*0/                        BLK00680
      DATA DAYMAX/20*-999.99/,MAXDAY/20*0/                               BLK00690
      END                                                                BLK00700
```

```
C     PROGRAM DIDOT
C     +--------------------------------------------------------------+
C     |                                                              |
C     |   *******************************************                |
C     |   *    D I D O T === M A I N   P R O G R A M  *               |
C     |   *    ---------    ----------------------    *               |
C     |   *******************************************                |
C     |                                                              |
C     +--------------------------------------------------------------+
C
C     THIS IS THE (DIDOT) MAIN PROGRAM WHICH DOES THE FOLLOWING THINGS:
C        1)   READS THE FIRST FEW CONTROL CARDS, AND PROCESSES THEM;
C        2)   READS THE INPUT PACKAGE NAME CARDS AND CALLS THE
C                  PROPER PACKAGE INPUT ROUTINE;
C        3)   CONTROLS THE OVERLAYING AND EXECUTION OF THE COMPUTATIONS;
C        4)   PREPARES AND CONTROLS THE PRINTED AND DISK/TAPE OUTPUTS.
C     NOTE THE STRUCTURE OF THE COMMON BLOCKS....SEE BLOCK DATA AND
C        THE USER DOCUMENTATION FOR DETAILS
      COMMON /ORIGIN/ XO,YO,UNIT NEAST,NNORTH,NPOLLS
      COMMON /RECEPS/ INDEX(225),INDPPT(20),KATCH(20),
     X                NEXAMS(10),KNTEX
      COMMON /DISKOP/ DAYDEP(2,225),PPTCON(2,20,24)
      COMMON /PTSRCS/ QXY(4,10),PARAMS(5,10), HTABL(3,10,10),
     X                DISTBL(3,10,10),COORDS(2,10),IFLAG(10),KNTPTS
      COMMON /ARSRCS/ VGRID,ACROSS,UP,QXYS(5,50),KNTARS
      COMMON /CITIES/ QXYCIT(5,5),SIGZO(6,5),XTENT(6,5),IFLAGC(5),KNTCTY
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)
      COMMON /IDENTS/ KEY1(6),DOLLAR
      COMMON /METDAT/ IDAYS(366),THETA(24),U(24),KST(24),HL(24),
     X                TEMP(24),IRAIN(25,5)
      COMMON /VALUES/ ADJWD(2),TAU(2),VDEPOS(2),
     X                VSETTL(2),SCAVNG(2),BKGRND(2),RUFLEN
      COMMON /CHITAB/ HRMAXP(2,5),HRMAXA(2,5),HRMAXC(2,5),
     X                MAXHRP(2,5,2),MAXHRA(2,5,2),MAXHRC(2,5,2),
     X                DAYMAX(2,10),MAXDAY(2,10)
C
C=== *******************************************************
      COMMON / TEST / IDBUG,LUNDBG
      LOGICAL IDBUG
C=== *******************************************************
C
      REAL*8 KEY1,KEND,A2,IMAGE(9),TITLIN(10),BLANKS
      DATA KEND/6HENDATA/,BLANKS/8H        /,IDUMMY,INPUT/2*-1/
      EQUIVALENCE (UNIT,GRID),(XO,XSWCOR),(YO,YSWCOR)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     NAMELIST /CPARMS/ TAU,VSETTL,VDEPOS,UNIT,XSWCOR,YSWCOR,RUFLEN,
C    X   SCAVNG,BKGRND
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      CALL TABSET
      WRITE (LUNPRT,600)
C
C=== *******************************************************
      READ (10,222,END=111,ERR=111) IDBUG
222   FORMAT (L1)
C=== *******************************************************
```

```
C                                                                       DID00580
  111 READ (LUNIN,555,END=6000) TITLIN                                  DID00590
  555 FORMAT(10A8)                                                      DID00600
      WRITE (LUNPRT,556) TITLIN                                         DID00610
  556 FORMAT(1H0,20X,10A8)                                              DID00620
      IF(TITLIN(1) .EQ. BLANKS)  GO TO 1                                DID00630
      GO TO 111                                                         DID00640
    1 CONTINUE                                                          DID00650
      READ(LUNIN,500,END=6000,ERR=9998) A1,A2,IMAGE,N                   DID00660
      IF( A1 .EQ. DOLLAR) GO TO 2                                       DID00670
 9998 WRITE(LUNPRT,601) A1,A2,IMAGE                                     DID00680
      GO TO 1                                                           DID00690
    2 IF(A2 .EQ. KEND) GO TO 1000                                       DID00700
      DO 3 L=1,6                                                        DID00710
      IF( A2 .EQ. KEY1(L)) GO TO (10,20,30,40,50,60), L                 DID00720
    3 CONTINUE                                                          DID00730
      WRITE(LUNPRT,602) A2                                              DID00740
      GO TO 1                                                           DID00750
   10 CALL PTSRCE(N)                                                    DID00760
      GO TO 1                                                           DID00770
   20 CALL ARSRCE(N,INPUT)                                              DID00780
      GO TO 1                                                           DID00790
   30 CALL CITY(N,INPUT)                                                DID00800
      GO TO 1                                                           DID00810
   40 CALL CATCHS(INPUT)                                                DID00820
      GO TO 1                                                           DID00830
   50 CALL METCTL(METPRT)                                               DID00840
      GO TO 1                                                           DID00850
C60   READ (LUNIN, CPARMS)                                              DID00860
   60 CONTINUE                                                          DID00870
      READ(LUNIN,501,END=6000)UNIT,XSWCOR,YSWCOR,RUFLEN                 DID00880
  501 FORMAT(10X,5F10.0)                                                DID00890
      READ(LUNIN,501,END=6000)TAU                                       DID00900
      READ(LUNIN,501,END=6000)VSETTL                                    DID00910
      READ(LUNIN,501,END=6000)VDEPOS                                    DID00920
      READ(LUNIN,501,END=6000)SCAVNG                                    DID00930
      READ(LUNIN,501,END=6000)BKGRND                                    DID00940
      INPUT=+1                                                          DID00950
      WRITE (LUNPRT,701)                                                DID00960
      WRITE (LUNPRT,702)  TAU                                           DID00970
      WRITE (LUNPRT,703)  VSETTL                                        DID00980
      WRITE (LUNPRT,704)  VDEPOS                                        DID00990
      WRITE (LUNPRT,705)  SCAVNG                                        DID01000
      WRITE (LUNPRT,706)  BKGRND                                        DID01010
      WRITE (LUNPRT,707)  RUFLEN                                        DID01020
      GO TO 1                                                           DID01030
C                                                                       DID01040
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DID01050
C                                                                       DID01060
  500 FORMAT( A2,A6,9A8,T9,I2)                                          DID01070
  600 FORMAT(1H1)                                                       DID01080
  601 FORMAT('0'/'0'/'0',10X,'ERROR IN INPUT...CARD CAN NOT BE IDENT',  DID01090
     X  'IFIED:'/23X,A2,A6,9A8//11X,'SCANNING FOR NEXT GOOD CARD.')     DID01100
  602 FORMAT('0'/'0'/'0',10X,'ERROR IN INPUT...NO SUCH INPUT PACKAGE ', DID01110
     X  'NAME AS:',A6,'.   SCANNING FOR NEXT GOOD CARD.')               DID01120
  701 FORMAT(1H1,50X,'P O L L U T A N T'/51X,1H1,9X,1H2/)               DID01130
  702 FORMAT(5X,'HALF-LIFE (HOUR)',T41,1P2G15.5,                        DID01140
     X         T90,'TAU'/)
 703  FORMAT(5X,'SETTLING VELOCITY (M/S)',T41,1P2G15.5,                 DID01150
     X         T90,'VSETTL'/)                                           DID01160
 704  FORMAT(5X,'DEPOSITION VELOCITY (M/S)',T41,1P2G15.5,               DID01170
     X         T90,'VDEPOS'/)                                           DID01180
 705  FORMAT(5X,'SCAVENGING RATIO ',T41,1P2G15.5,                       DID01190
     X         T90,'SCAVNG'/)                                           DID01200
 706  FORMAT(5X,'BACKGROUND CONC. (UG/M**3)',T41,1P2G15.5,              DID01210
     X         T90,'BKGRND'/)                                           DID01220
 707  FORMAT(1H0/1H0,5X,'ROUGHNESS LENGTH (METERS)',F10.3,              DID01230
     X         T90,'RUFLEN'/)                                           DID01240
C                                                                       DID01250
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  DID01260
C                                                                       DID01270
1000  IF(INPUT)  6000,6000,1001                                         DID01280
1001  IF(KNTPTS+KNTARS+KNTCTY.EQ.0)GO TO 6000                           DID01290
      NRECEP=NNORTH*NEAST                                               DID01300
      VGRID=1.0/UNIT                                                    DID01310
      ACROSS=FLOAT(NEAST)                                               DID01320
      UP=FLOAT(NNORTH)                                                  DID01330
      IF (KNTPTS .GT. 0) CALL PTLIST(IDUMMY)                            DID01340
      IF (KNTARS .GT. 0) CALL ARLIST(IDUMMY,INPUT)                      DID01350
      IF (KNTCTY .GT. 0) CALL CTLIST(IDUMMY,INPUT)                      DID01360
      CALL KPOLLS                                                       DID01370
      CALL WRITE1 (IDUMMY)                                              DID01380
C                                                                       DID01390
C                                                                       DID01400
C                                                                       DID01410
C                                                                       DID01420
1999  CONTINUE                                                          DID01430
2000  IOK=0                                                             DID01440
      CALL GETMET(METPRT,IOK,IRON)                                      DID01450
      IF(IOK)2000,2010,5000                                             DID01460
2010  CALL PRESET                                                       DID01470
      IF (KNTPTS .EQ. 0) GO TO 2200                                     DID01480
      KHR=0                                                             DID01490
2001  KHR=KHR+1                                                         DID01500
      IF(KHR .GT. 24)GO TO 2200                                         DID01510
      CALL HOURLY (T,TRAD,W,TA,HMIX,ISTAB,KHR)                          DID01520
C                                                                       DID01530
      CALL DUMMY                                                        DID01540
C                                                                       DID01550
C                                                                       DID01560
      DO 2150 KREC=1,NRECEP                                             DID01570
      CALL XYLOC (XREC,YREC,KREC,ICOL,IROW)                             DID01580
2020  CALL PTGEOM(XREC,YREC,KREC,KHR,ISTAB,IKST,IWIND,HMIX,W,TRAD)      DID01590
C                                                                       DID01600
      CALL PTCONC(W,TA,HMIX,ISTAB,IKST,IWIND,INDEX(KREC),KREC,KHR)      DID01610
C                                                                       DID01620
C                                                                       DID01630
2150  CONTINUE                                                          DID01640
      GO TO 2001                                                        DID01650
C                                                                       DID01660
C                                                                       DID01670
C                                                                       DID01680
2200  IF(KNTARS.EQ.0)GO TO 2300                                         DID01690
C                                                                       DID01700
                                                                        DID01710
```

```
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DID01720
      KHR = 0                                                                         DID01730
2201  KHR = KHR + 1                                                                   DID01740
      IF (KHR .GT. 24) GO TO 2300                                                     DID01750
      IF (KHR .EQ. 1) CALL ARGEOM                                                     DID01760
      CALL HOURLY (T,TRAD,W,TA,HMIX,ISTAB,KHR)                                        DID01770
      CALL ARCONC (KHR,ISTAB,W)                                                       DID01780
      GO TO 2201                                                                      DID01790
C                                                                                     DID01800
C * * * * *  ********                                                                 DID01810
2300  IF(KNTCTY .EQ. 0) GO TO 3000                                                    DID01820
      KHR = 0                                                                         DID01830
2301  KHR = KHR + 1                                                                   DID01840
      IF (KHR .GT. 24) GO TO 3000                                                     DID01850
      CALL HOURLY (T,TRAD,W,TA,HMIX,ISTAB,KHR)                                        DID01860
      CALL DUMMY                                                                      DID01870
      CALL CTGEOM (TRAD,ISTAB)                                                        DID01880
      DO 2450 KCTY = 1,KNTCTY                                                         DID01890
      IFL = IFLAGC(KCTY)                                                              DID01900
      IF (IFL .EQ. -1) GO TO 2450                                                     DID01910
      DO 2400 KR = 1,NRECEP                                                           DID01920
      CALL XYLOC (XR,YR,KR,ICOL,IROW)                                                 DID01930
      CALL CTCONC(HMIX,TRAD,W,XR,YR,KR,INDEX(KR),KCTY,KHR,ISTAB,IFL)                  DID01940
2400  CONTINUE                                                                        DID01950
2450  CONTINUE                                                                        DID01960
      GO TO 2301                                                                      DID01970
C                                                                                     DID01980
C                                                                                     DID01990
C                                                                                     DID02000
3000  CONTINUE                                                                        DID02010
C                                                                                     DID02020
3999  IF (.NOT. IDBUG) GO TO 4000                                                     DID02030
C=== ************************************************************                     DID02040
C===  PUT DEBUG PRINTS HERE.....                                                      DID02050
C=== ************************************************************                     DID02060
C                                                                                     DID02070
4000  CALL DAYEND(IRON)                                                               DID02080
      GO TO 2000                                                                      DID02090
5000  CONTINUE                                                                        DID02100
C                                                                                     DID02110
5199  IF (.NOT. IDBUG) GO TO 5500                                                     DID02120
C=== ************************************************************                     DID02130
C===  PUT DEBUG PRINTS HERE.....                                                      DID02140
C=== ************************************************************                     DID02150
C                                                                                     DID02160
5500  CALL WRAPUP(NPOLLS,KNTPTS,KNTARS,KNTCTY)                                        DID02170
6000  CONTINUE                                                                        DID02180
      STOP                                                                            DID02190
      END                                                                             DID02200
      SUBROUTINE TABSET
C     ROUTINE TO INITIALIZE TABLES
      COMMON /TAB001/ HLOW,HHIGH,HDEL,XLTAB(50,4),
     X                XLOW,XHIGH,XDEL,SYTAB(50,4),SZTAB(50,4),
     X                ELOW,EMAX,EDEL,ETAB(100),NCLASS,NHT,NXC,NEXP
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,
     X                T,P,SYCOEF(4),SYEXP(4),SZCOEF(4),SZEXP(4),
     X                R2PI,ROOT2,ROOTPI,
     X                J1STAB(7),J2STAB(7)
C
C
C     +----------------------------------------------------------------+
C     |                                                                |
C     |  PICK ONE OF THE FOLLOWING LINES: (FIRST LINE FOR IBM;         |
C     |                                    SECOND LINE FOR PDP-11)     |
C     |  ***********************************************               |
C     |                                                                |
        CALL ERRSET (208,1000,-1,1,0)
        CALL ERRSET (212,1000,-1,1,0,1)
C     |                                                                |
C     | CALL ERRSET (74,T,F,F,F,30000)                                 |
C     |                                                                |
C     |  ***********************************************               |
C     |                                                                |
C     +----------------------------------------------------------------+
C
      DO 50 IC=1,NCLASS
      DO 10 IH=1,NHT
      H=HLOW+(IH-1)*HDEL
      XLTAB(IH,IC)=((0.47*H)/SZCOEF(IC))**(1.0/SZEXP(IC))
10    CONTINUE
C     BUILD TABLE FOR XL S.T. SZ(XL) = 0.47*H
      HHIGH=H
      DO 20 IX=1,NXC
      DIST=XLOW+(IX-1)*XDEL
      SYTAB(IX,IC)=SYCOEF(IC)*DIST**SYEXP(IC)
      SZTAB(IX,IC)=SZCOEF(IC)*DIST**SZEXP(IC)
20    CONTINUE
      XHIGH=DIST
C     BUILD TABLE FOR SY AND SZ
50    CONTINUE
C     ALL DONE WITH TABLES DEPENDING ON STABILITY CLASS
      DO 60 IE=1,NEXP
      ARG=ELOW+(IE-1)*EDEL
      ETAB(IE)=EXP(-.5*ARG**2)
60    CONTINUE
      EMAX=ARG
C     BUILD TABLE FOR EXP(-.5*X**2)
C
      RETURN
      END
      SUBROUTINE PTSRCE(NPOINT)                                         PTS00010
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,                       PTS00020
     X                T,P,YCOEFF(4),YPOWER(4),ZCOEFF(4),ZPOWER(4),      PTS00030
     X                R2PI,ROOT2,ROOTPI,                                PTS00040
     X                J1STAB(7),J2STAB(7)                               PTS00050
      COMMON /PTSRCS/ QXY(4,10),PARAMS(5,10),HTABL(3,10,10),            PTS00060
     X                DISTBL(3,10,10),COORDS(2,10),IFLAG(10),KNTPTS     PTS00070
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)  PTS00080
C                                                                       PTS00090
C=== ************************************************************        PTS00100
      COMMON / TEST / IDBUG,LUNDBG                                      PTS00110
      LOGICAL IDBUG                                                     PTS00120
C=== ************************************************************        PTS00130
C                                                                       PTS00140
      REAL      NAMES(2,10)                                             PTS00150
      EQUIVALENCE (TEMPK,TS),(VOLFLX,VF),(VELOC,VS),(DIAM,DS)           PTS00160
      EQUIVALENCE (HEIGHT,HS),(BOUYFX,F)                                PTS00170
C                                                                       PTS00180
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * PTS00190
C                                                                       PTS00200
 7000 N=NPOINT                                                          PTS00210
      IF(NPOINT-KNTPTS .NE. 1) GO TO 1000                               PTS00220
      IF(NPOINT .GT. 10) GO TO 1010                                     PTS00230
      READ (LUNIN,500,END=1020,ERR=9998) (NAMES(J,NPOINT),J=1,2),       PTS00240
     X    Q1,Q2,X,Y,HEIGHT,DIAM,VELOC,TEMPK,VOLFLX                      PTS00250
  500 FORMAT(2X,2A4,2F10.0,10X,2F10.0/10X,5F10.0)                       PTS00260
      IF (Q1+Q2 .EQ. 0.0) GO TO 993                                     PTS00270
      DELT=TS-T                                                         PTS00280
      IF(VF)99,2,11                                                     PTS00290
    2 IF(VS)99,98,3                                                     PTS00300
    3 IF(DS)99,98,4                                                     PTS00310
    4 VF=PION4*DS*DS*VS                                                 PTS00320
      GO TO 50                                                          PTS00330
   11 IF(VS)99,12,21                                                    PTS00340
   12 IF(DS)99,98,13                                                    PTS00350
   13 VS=VF/(PION4*DS*DS)                                               PTS00360
      GO TO 50                                                          PTS00370
   21 DS=SQRT(VF/(VS*PION4))                                            PTS00380
   50 BOUYFX=3.12139*VF*DELT/TS                                         PTS00390
      GO TO 1500                                                        PTS00400
   98 WRITE(LUNPRT,981)                                                 PTS00410
      GO TO 993                                                         PTS00420
   99 WRITE(LUNPRT,991)                                                 PTS00430
      GO TO 993                                                         PTS00440
 1000 WRITE(LUNPRT,992) KNTPTS,NPOINT                                   PTS00450
      GOTO 993                                                          PTS00460
 1010 WRITE(LUNPRT,982) NPOINT                                          PTS00470
      GO TO 993                                                         PTS00480
 1020 WRITE(LUNPRT,983)                                                 PTS00490
      GO TO 993                                                         PTS00500
 9998 WRITE(LUNPRT,984)                                                 PTS00510
      GO TO 993                                                         PTS00520
  981 FORMAT('OTOO MANY MISSING PARAMS. IN STACK DATA'//)               PTS00530
  982 FORMAT('ONUMBER OF THIS POINT SOURCE EXCEEDS DIMENSIONED SPACE...'PTS00540
     X,' MAXIMUM ALLOWED=10, THIS ONE IS NUMBER ',I3,1H.//)             PTS00550
  983 FORMAT('OUNANTICIPATED END-OF-INPUT....'//)                       PTS00560
  984 FORMAT('OERROR ON READING INPUT...UNINTELLIGIBLE CONVERSION'//)   PTS00570
  991 FORMAT('ONEGATIVE PARAMETER VALUE IN STACK DATA'//)               PTS00580
  992 FORMAT('OPOINT SOURCE INPUT OUT OF SEQUENCE...PREVIOUS INDEX=',   PTS00590
     X   I3,', THIS POINT SOURCE INDEX=',I3,1H.//)                      PTS00600
  993 RETURN                                                            PTS00610
 1500 CONTINUE                                                          PTS00620
      QXY(1,N)=Q1                                                       PTS00630
      QXY(2,N)=Q2                                                       PTS00640
      QXY(3,N)=X                                                        PTS00650
      QXY(4,N)=Y                                                        PTS00660
      PARAMS(1,N)=HS                                                    PTS00670
      PARAMS(2,N)=DS                                                    PTS00680
      PARAMS(3,N)=VS                                                    PTS00690
      PARAMS(4,N)=TS                                                    PTS00700
      PARAMS(5,N)=VF                                                    PTS00710
C     BETWEEN HERE AND STATEMENT 20, DO CALCULATIONS FOR UNSTABLE + NEUTRPTS00720
7     IF(F-55.)8,9,9                                                    PTS00730
8     XST=14.*F**.625                                                   PTS00740
      GO TO 10                                                          PTS00750
9     XST=34.*F**.4                                                     PTS00760
10    DISTF=3.5*XST                                                     PTS00770
      DO 19 IWS=1,10                                                    PTS00780
      U=WSPDS(IWS)                                                      PTS00790
      DELHF=1.6*F**0.333333*DISTF**0.666667/U                           PTS00800
      DISTBL(1,IWS,N)=1.0E-3*DISTF                                      PTS00810
19    HTABL(1,IWS,N)=HS+DELHF                                           PTS00820
   20 DTHDZ = 0.02                                                      PTS00830
      IKST=2                                                            PTS00840
      GO TO 24                                                          PTS00850
   23 DTHDZ = 0.035                                                     PTS00860
      IKST=3                                                            PTS00870
   24 S = G*DTHDZ/T                                                     PTS00880
      DO 30 IWS=1,10                                                    PTS00890
      U=WSPDS(IWS)                                                      PTS00900
      DHA = 2.6*(F/(U*S))**0.333333                                     PTS00910
      DELHF = 5.0*F**0.25/S**0.375                                      PTS00920
      IF(DHA-DELHF) 25,25,27                                            PTS00930
   25 DELHF = DHA                                                       PTS00940
   27 DISTF = 3.14159*U/S**0.5                                          PTS00950
      DISTBL(IKST,IWS,N)=1.0E-3*DISTF                                   PTS00960
      HTABL(IKST,IWS,N)=HS+DELHF                                        PTS00970
30    CONTINUE                                                          PTS00980
      IF(IKST .EQ. 2) GO TO 23                                          PTS00990
      KNTPTS=KNTPTS+1                                                   PTS01000
      RETURN                                                            PTS01010
C                                                                       PTS01020
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * PTS01030
C                                                                       PTS01040
      ENTRY PTLIST                                                      PTS01050
      WRITE (LUNPRT,700)                                                PTS01060
      DO 1001 J=1,KNTPTS                                                PTS01070
 1001 WRITE (LUNPRT,701) J,NAMES(1,J),NAMES(2,J),(QXY(I,J),I=1,4),      PTS01080
     X       (PARAMS(I,J),I=1,5)                                        PTS01090
  700 FORMAT(1H1///31X,'POINT SOURCE INPUT DATA'/                       PTS01100
     X    T3,'POINT ',T10,'NAME',T20,'E-RATE1',T30,'E-RATE2',T50,       PTS01110
     X    'X-COORD.',                                                   PTS01120
     X      T60,'Y-COORD.',T70,'STK.HGT',T80,'DIAM',T90,'VELOC',        PTS01130
     X      T100,'TEMP-K',T110,'VOLFLUX'//)                             PTS01140
```

```
701    FORMAT(3X,I3,T8,2A4,2X,1P2G10.3,10X,0P2F10.3,1P5G10.3/)
C
100    IF (.NOT. IDBUG) RETURN
C=== ***********************************************************
C===   PUT DEBUG PRINTS HERE.....
C      DO 125 K1=1,3
C      WRITE (LUNDBG,800)  K1,(K2,WSPDS(K2),K2=1,10)
C      DO 124 K3=1,KNTPTS
C      WRITE (LUNDBG,801) K3,(HTABL(K1,K2,K3),K2=1,10),
C     X                       (DISTBL(K1,K2,K3),K2=1,10)
C 124    CONTINUE
C 125    CONTINUE
C 800    FORMAT (1H0/1H0/3X,'**** PTSRCE TABLE FOR STABILITY GROUP: ',I3/
C       X         /3X,'SRCE #  SPDS= ',10(I2,1H=,F7.3)/)
C 801    FORMAT (5X,I2,10X,1P10G10.2/17X,10G10.2/)
C=== ***********************************************************
C
       RETURN
       END
```

```
      SUBROUTINE ARSRCE(NAREA,INPUT)
      COMMON /ARSRCS/  VGRID,ACROSS,UP,QXYS(5,50),KNTARS                ARS00010
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)  ARS00020
      COMMON /ORIGIN/ XSWCOR,YSWCOR,UNIT,NEAST,NNORTH                   ARS00030
      REAL      QRATE(2),INAMES(2,50)                                   ARS00040
      IF( INPUT)  993,993,7000                                          ARS00050
7000  CONTINUE                                                          ARS00060
      IF(UNIT)1030,1030,1                                               ARS00070
1     IF(NAREA-KNTARS .NE. 1) GO TO 1000                                ARS00080
      IF(NAREA .GT. 50) GO TO 1010                                      ARS00090
      READ(LUNIN,500,END=1020,ERR=9998) (INAMES(J,NAREA),J=1,2),        ARS00100
     X  Q1,Q2,XCOR,YCOR,SIDE                                            ARS00110
500   FORMAT(2X,2A4,2F10.0,10X,3F10.0)                                  ARS00120
      IF(SIDE)9998,10,11                                                ARS00130
10    SIDE=UNIT                                                         ARS00140
11    XDISP=(XCOR-XSWCOR)/UNIT                                          ARS00150
      YDISP=(YCOR-YSWCOR)/UNIT                                          ARS00160
      IF(ABS(XDISP-IFIX(XDISP)) .GT. 0.001) GO TO 1030                  ARS00170
      IF(ABS(YDISP-IFIX(YDISP)) .GT. 0.001) GO TO 1030                  ARS00180
      IF(AMOD(SIDE,UNIT) .NE. 0.0) GO TO 1030                           ARS00190
      KNTARS=KNTARS+1                                                   ARS00200
      QXYS(1,KNTARS)=Q1                                                 ARS00210
      QXYS(2,KNTARS)=Q2                                                 ARS00220
      QXYS(3,KNTARS)=XCOR                                               ARS00230
      QXYS(4,KNTARS)=YCOR                                               ARS00240
      QXYS(5,KNTARS)=SIDE                                               ARS00250
      RETURN                                                            ARS00260
      ENTRY ARLIST                                                      ARS00270
      IF( INPUT)  993,993,7001                                          ARS00280
7001  WRITE(LUNPRT,700)                                                 ARS00290
      DO 2001 J=1,KNTARS                                                ARS00300
      SIDE=QXYS(5,J)                                                    ARS00310
      AREA=SIDE*SIDE                                                    ARS00320
      DO 20 L=1,2                                                       ARS00330
20    QRATE(L)=QXYS(L,J)/AREA                                           ARS00340
2001  WRITE(LUNPRT,701) J,(INAMES(I,J),I=1,2),(QXYS(I,J),I=1,5),        ARS00350
     X     AREA,QRATE                                                   ARS00360
      RETURN                                                            ARS00370
700   FORMAT(1H1///20X,'AREA SOURCE INPUT AND COMPUTED DATA'//          ARS00380
     X   3X,'NO.',5X,'NAME',6X,'E-RATE (G/S)',12X,'CORNER LOC. (KM)',   ARS00390
     X   2X,'WIDTH (KM)',2X,'AREA (KM**2)',2X,'E-RATE (G/KM**2/S)'/)    ARS00400
701   FORMAT(2X,I3,6X,2A4,2X,1P2G10.3,0P4F10.3,1P2G10.3)                ARS00410
1000  WRITE(LUNPRT,992) NAREA,KNTARS                                    ARS00420
      GOTO 993                                                          ARS00430
1010  WRITE(LUNPRT,982) KNTARS                                          ARS00440
      GO TO 993                                                         ARS00450
1020  WRITE(LUNPRT,983)                                                 ARS00460
      GO TO 993                                                         ARS00470
1030  WRITE(LUNPRT,991)                                                 ARS00480
      GO TO 993                                                         ARS00490
9998  WRITE(LUNPRT,984)                                                 ARS00500
      GO TO 993                                                         ARS00510
982   FORMAT('0NUMBER OF THIS AREA SOURCE EXCEEDS DIMENSIONED SPACE...' ARS00520
     X,' MAXIMUM ALLOWED=50, THIS ONE IS NUMBER ',I3,1H.//)             ARS00530
983   FORMAT('0UNANTICIPATED END-OF-INPUT....'//)                       ARS00540
984   FORMAT('0ERROR ON READING INPUT...UNINTELLIGIBLE CONVERSION'//)   ARS00550
991   FORMAT('0THIS AREA SOURCE IS NOT CONSISTENT WITH RECEPTOR GRID ', ARS00560
     X    'SYSTEM.'/)                                                   ARS00580
 992  FORMAT('0AREA SOURCE INPUT OUT OF SEQUENCE...PREVIOUS INDEX=',    ARS00590
     X    I3,'  THIS AREA SOURCE INDEX=',I3,1H.//)                      ARS00600
 993  STOP 43                                                           ARS00610
      END                                                               ARS00620
      SUBROUTINE CITY(NCITY,INPUT)
      COMMON /CITIES/ QXYCIT(5,5),SIGZ0(6,5),XTENT(6,5),IFLAGC(5),KNTCTY
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)
      COMMON /ORIGIN/ X0,Y0,UNIT,NEAST,NNORTH,NPOLLS
C
      REAL      QRATE(2),JNAMES(2,5),SZ0(6)
C * * * * * * * * * * *
      IF( INPUT)  981,981,7000
  981 WRITE (LUNPRT,781)
  781 FORMAT(1H0,'PROGRAM SHUT DOWN BECAUSE CITY-SOURCE INPUT CANNOT BE
     X PROCESSED BEFORE MAIN PROGRAM CONTROL PARAMETERS (CPARMS) ARE ',
     X 'GIVEN.'/)
      STOP 781
 7000 CONTINUE
      IF(UNIT)982,982,1
  982 WRITE (LUNPRT,782)
  782 FORMAT(1H0,'PROGRAM SHUT DOWN BECAUSE CITY-SOURCE INPUT CANNOT BE
     X PROCESSED BEFORE A POSITIVE VALUE OF THE UNIT PARAMETER IS',
     X 'GIVEN.'/)
      STOP 782
    1 IF(NCITY-KNTCTY .NE. 1) GO TO 1000
      IF(NCITY .GT. 5) GO TO 1010
      READ(LUNIN,500,END=1020,ERR=1040) (JNAMES(J,NCITY),J=1,2),
     X  Q1,Q2,XCOR,YCOR,SIDE,(SZ0(L),L=1,6)
  500 FORMAT(2X,2A4,2F10.0,10X,3F10.0/10X,6F10.0)
      IF(SIDE)987,10,11
  987 WRITE(LUNPRT,787) SIDE
  787 FORMAT(1H0,'PROGRAM SHUT DOWN BECAUSE CITY-SOURCE SIDE-LENGTH IS'
     X,'NEGATIVE AND =',F10.4)
      STOP 787
   10 SIDE=10.0*UNIT
   11 IF (SIDE .LT. 5.0 .OR. SIDE .GT. 75.0) GO TO 1030
      XDISP=(XCOR-X0)/UNIT
      YDISP=(YCOR-Y0)/UNIT
      IF(ABS(XDISP-IFIX(XDISP)) .GT. 0.001) GO TO 1030
      IF(ABS(YDISP-IFIX(YDISP)) .GT. 0.001) GO TO 1030
      IF(AMOD(SIDE,UNIT) .NE. 0.0) GO TO 1030
      KNTCTY=KNTCTY+1
      QXYCIT(1,KNTCTY)=Q1
      QXYCIT(2,KNTCTY)=Q2
      QXYCIT(3,KNTCTY)=XCOR
      QXYCIT(4,KNTCTY)=YCOR
      QXYCIT(5,KNTCTY)=SIDE
      IF(SZ0(1) .LT. 25.0) SZ0(1) = 25.0
      SIGZ0(1,KNTCTY) = SZ0(1)
      DO 12 K = 2,6
      SIGZ0(K,KNTCTY) = SZ0(K)
      IF (SZ0(K) .EQ. 0.0) SIGZ0(K,KNTCTY) = SIGZ0(1,KNTCTY)
   12 CONTINUE
      RETURN
C
      ENTRY CTLIST
      IF( INPUT)  981,981,7001
 7001 WRITE(LUNPRT,700)
      DO 2001 J=1,KNTCTY
      SIDE=QXYCIT(5,J)
      CTYSIZ=SIDE*SIDE
      DO 20 L=1,2                                                      NUD06210
   20 QRATE(L)=QXYCIT(L,J)/CTYSIZ                                      NUD06220
      WRITE(LUNPRT,701) J,(JNAMES(I,J),I=1,2),(QXYCIT(I,J),I=1,5),     NUD06230
     X     CTYSIZ,QRATE,(SIGZO(K,J),K=1,6)                             NUD06240
      QXYCIT(3,J) = QXYCIT(3,J) + SIDE/2.0                             NUD06250
      QXYCIT(4,J) = QXYCIT(4,J) + SIDE/2.0                             NUD06260
 2001 CONTINUE                                                         NUD06270
      RETURN                                                           NUD06280
  700 FORMAT(1H1///20X,'CITY SOURCE INPUT AND COMPUTED DATA'//         NUD06290
     X   3X,'NO.',5X,'NAME',6X,'E-RATE (G/S)',12X,'CORNER LOC. (KM)',  NUD06300
     X   2X,'WIDTH (KM)',2X,'CITY (KM**2)',2X,'E-RATE (G/KM**2/S)'/)   NUD06310
  701 FORMAT(2X,I3,6X,2A4,2X,1P2G10.3,0P4F10.3,1P2G10.3/               NUD06320
     X   11X,'SZO=',0P6F6.2//)                                         NUD06330
 1000 WRITE(LUNPRT,992) NCITY,KNTCTY                                   NUD06340
      GOTO 993                                                         NUD06350
 1010 WRITE(LUNPRT,882) KNTCTY                                         NUD06360
      GO TO 993                                                        NUD06370
 1020 WRITE(LUNPRT,983)                                                NUD06380
      GO TO 993                                                        NUD06390
 1030 WRITE(LUNPRT,991)                                                NUD06400
      GO TO 993                                                        NUD06410
 1040 WRITE(LUNPRT,984)                                                NUD06420
      GO TO 993                                                        NUD06430
  882 FORMAT('0NUMBER OF THIS CITY SOURCE EXCEEDS DIMENSIONED SPACE...' NUD06440
     X,' MAXIMUM ALLOWED=5, THIS ONE IS NUMBER ',I3,1H.//)             NUD06450
  983 FORMAT('0UNANTICIPATED END-OF-INPUT....'//)                      NUD06460
  984 FORMAT('0ERROR ON READING INPUT...UNINTELLIGIBLE CONVERSION'//)  NUD06470
  991 FORMAT('0THIS CITY SOURCE IS NOT CONSISTENT WITH RECEPTOR GRID ', NUD06480
     X   'SYSTEM.'/)                                                   NUD06490
  992 FORMAT('0CITY SOURCE INPUT OUT OF SEQUENCE...PREVIOUS INDEX=',   NUD06500
     X   I3,',  THIS CITY SOURCE INDEX=',I3,1H.//)                     NUD06510
  993 STOP 43                                                          NUD06520
      END                                                              NUD06530
      SUBROUTINE CATCHS (INPUT)                                         CAT00010
      COMMON /RECEPS/ INDEX(225),INDPPT(20),KATCH(20),                  CAT00020
     X                NEXAMS(10),KNTEX                                  CAT00030
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)  CAT00040
      COMMON /ORIGIN/   XSWCOR,YSWCOR,UNIT,NEAST,NNORTH,NPOLLS          CAT00050
      DIMENSION LINE(25),INDJ(3),KATUNS(20,25)                          CAT00060
      EQUIVALENCE (J1,INDJ(1)),(J2,INDJ(2)),(J3,INDJ(3))                CAT00070
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CAT00080
C     NAMELIST /MAPIN/ INDEX,NNORTH,NEAST,NEXAMS,INDPPT,KNTEX           CAT00090
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CAT00100
      IF( INPUT) 100,100,1                                              CAT00110
C1    READ(LUNIN,MAPIN)                                                 CAT00120
1     READ(LUNIN,501,END=100) NNORTH,NEAST                              CAT00130
501   FORMAT(10X,10I5)                                                  CAT00140
      MAX=NNORTH*NEAST                                                  CAT00150
      IF(MAX .GT. 225) GOTO100                                          CAT00160
      READ(LUNIN, * ,END=100)(INDEX(J),J=1,MAX)                         CAT00170
      READ(LUNIN,501,END=100) KNTEX                                     CAT00180
      READ(LUNIN, * ,END=100) (NEXAMS(J),J=1,KNTEX)                     CAT00190
      READ(LUNIN,501,END=100) INDPPT                                    CAT00200
      WRITE (LUNPRT,601) UNIT,NNORTH,NEAST,XSWCOR,YSWCOR                CAT00210
601   FORMAT('1   UNIT SQUARE SIZE= ',F6.2,                             CAT00220
     X    ' KM.,  NUMBER OF UNIT SQUARES ',                             CAT00230
     X    'TO THE NORTH= ',I4,', NUMBER OF UNIT SQUARES TO THE EAST= ', CAT00240
     X      I4/'0   THE COORDINATES (X,Y) OF THE SOUTHWEST CORNER= ',   CAT00250
     X    1H(,F8.3,1H,F8.3,1H)///T13,'KNTR',T22,'NCOL',T32,'NROW',      CAT00260
     X     T39,'BASIN',T52,                                             CAT00270
     X     'X-COORD',T71,'Y-COORD',T85,'RAIN-GAGE INDEX'//)             CAT00280
      MAX=NNORTH*NEAST                                                  CAT00290
      IF(MAX .GT. 225) GOTO100                                          CAT00300
C                                                                       CAT00310
C                                                                       CAT00320
      DO 2 J=1,MAX                                                      CAT00330
      CALL XYLOC (X,Y.J,NCOL,NROW)                                      CAT00340
      LJ=INDEX(J)                                                       CAT00350
      CALL INDXCK (LJ,N,J1,J2,J3,IFLAG)                                 CAT00360
C                                                                       CAT00370
      DO 40 KN = 1,N                                                    CAT00380
      IDKAT = INDJ(KN)                                                  CAT00390
      IF (IDKAT .LE. 20 ) GO TO 32                                      CAT00400
      WRITE(LUNPRT,605) J,LJ,KN,IDKAT                                   CAT00410
      STOP 32                                                           CAT00420
32    NEXT = KATCH(IDKAT) + 1                                           CAT00430
      IF (NEXT .LE. 25) GO TO 33                                        CAT00440
      IFLAG = -1                                                        CAT00450
      GO TO 34                                                          CAT00460
33    KATCH(IDKAT) = NEXT                                               CAT00470
      KATUNS(IDKAT,NEXT) = J                                            CAT00480
40    CONTINUE                                                          CAT00490
C                                                                       CAT00500
34    WRITE (LUNPRT,602) J,NCOL,NROW,LJ,X,Y,                            CAT00510
     X    (INDPPT(INDJ(KN)),KN=1,N)                                     CAT00520
      IF (IFLAG) 39,2,39                                                CAT00530
39    STOP 13                                                           CAT00540
2     CONTINUE                                                          CAT00550
602   FORMAT(T11,I4,T21,I4,T31,I4,T37,I6,T51,F8.3,T71,F8.3,             CAT00560
     X    T89,3I3)                                                      CAT00570
```

```
      WRITE(LUNPRT,600)                                                 CAT00580
C                                                                       CAT00590
C                                                                       CAT00600
600   FORMAT(1H1)                                                       CAT00610
605   FORMAT(1H0,10X,'CATCHMENT INDEX IS TOO LARGE...THIS VERSION OF ', CAT00620
     X   'DIDOT ALLOWS UP TO 20 CATCHMENTS.'/11X,'RECEPTOR (MAP SQUARE) CAT00630
     X NUMBER:',I4,', INPUT INDEX VALUE:',I6,'.  CATCHMENT NUMBER ',I2, CAT00640
     X ' FOR THIS SQUARE IS:',I2,1H./)                                  CAT00650
      WRITE (LUNPRT,611)                                                CAT00660
611   FORMAT (1H0//10X,'C R O S S - R E F E R E N C E   M A P :',       CAT00670
     X        ' * * * UNIT-SQUARES TO CATCHMENTS ***'//)                CAT00680
      DO 4 L=1,NNORTH                                                   CAT00690
      M=NNORTH-L+1                                                      CAT00700
      DO 3 J=1,NEAST                                                    CAT00710
      K=M+(J-1)*NNORTH                                                  CAT00720
3     LINE(J)=INDEX(K)                                                  CAT00730
4     WRITE(LUNPRT,603) (LINE(J),J=1,NEAST)                             CAT00740
603   FORMAT(1H0,2X,18I7/5X,7I7)                                        CAT00750
      WRITE(LUNPRT,600)                                                 CAT00760
      WRITE (LUNPRT,612)                                                CAT00770
612   FORMAT (1H0//10X,'C R O S S - R E F E R E N C E   M A P :',       CAT00780
     X        ' * * * UNIT-SQUARES TO RAIN-GAGES ***'//)                CAT00790
      DO 6 L=1,NNORTH                                                   CAT00800
      M=NNORTH-L+1                                                      CAT00810
      DO 5 J=1,NEAST                                                    CAT00820
      K=M+(J-1)*NNORTH                                                  CAT00830
C                                                                       CAT00840
      CALL INDXER (INDEX(K),N,J1,J2,J3)                                 CAT00850
C                                                                       CAT00860
      LXX=0                                                             CAT00870
      DO 41 LX=1,N                                                      CAT00880
41    LXX=100*LXX + INDPPT(INDJ(LX))                                    CAT00890
5     LINE(J)=LXX                                                       CAT00900
6     WRITE(LUNPRT,603) (LINE(J),J=1,NEAST)                             CAT00910
      WRITE (LUNPRT,600)                                                CAT00920
C                                                                       CAT00930
      DO 7 LL=1,20                                                      CAT00940
      KAREA=KATCH(LL)                                                   CAT00950
      IF (KAREA .EQ. 0) GO TO 7                                         CAT00960
      WRITE (LUNPRT,604) LL,KAREA,(KATUNS(LL,JJ),JJ=1,KAREA)            CAT00970
7     CONTINUE                                                          CAT00980
C                                                                       CAT00990
      IF(KNTEX .GT. 0) WRITE (LUNPRT,607) KNTEX,(NEXAMS(J),J=1,KNTEX)   CAT01000
C                                                                       CAT01010
604   FORMAT(3X,'CATCHMENT ',I3,' IS ',I4,' UNIT SQUARES. THEY ARE:',   CAT01020
     X   T61,15I4/T61,10I4)                                             CAT01030
607   FORMAT(3X,'THE FOLLOWING ',I2,' CATCHMENT ID''S ARE EXAMS WATER-',CAT01040
     X 'BODIES OR WATER-BODY CELLS:',5X,10I4)                           CAT01050
      RETURN                                                            CAT01060
C                                                                       CAT01070
      ENTRY WRITE1                                                      CAT01080
C                                                                       CAT01090
      KRTYPE = 1                                                        CAT01100
      WRITE (LUNNPS) KRTYPE                                             CAT01110
      DO 500 KDUMMY = 1,20                                              CAT01120
      LINE (KDUMMY) = 0                                                 CAT01130
      IF (KATCH (KDUMMY) .EQ. 0 ) GO TO 500                             CAT01140
      LINE (KDUMMY) = 1                                                 CAT01150
  500 CONTINUE                                                          CAT01160
      IF (KNTEX .EQ. 0 ) GO TO 555                                      CAT01170
      DO 550 KDUMMY = 1,KNTEX                                           CAT01180
      LINE (NEXAMS (KDUMMY) ) = 2                                       CAT01190
  550 CONTINUE                                                          CAT01200
  555 WRITE (LUNNPS) NPOLLS,(LINE(KDUMMY),KDUMMY=1,20)                  CAT01210
      RETURN                                                            CAT01220
  100 STOP                                                              CAT01230
      END                                                               CAT01240
      SUBROUTINE METCTL(METOPT)                                         NUD16220
      COMMON /METDAT/ IDAYS(366),THETA(24),U(24),KST(24),HL(24),        NUD16230
     X                TEMP(24),IRAIN(25,5),TSNOW                        NUD16240
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)  NUD16250
C                                                                       NUD16260
C     NAMELIST /MPARMS/ IDAYS,NPPT,LUNPPT,METPRT,TSNOW                  NUD16270
C                                                                       NUD16280
      METPRT=0                                                          NUD16290
      TSNOW = 32.0                                                      NUD16300
C     READ(LUNIN,MPARMS)                                                NUD16310
        READ (LUNIN,501,END=6000) IDAYS                                 NUD16320
  501   FORMAT (10X,3112/10X,2912/10X,3112/10X,3012/                    NUD16330
     X          10X,3112/10X,3012/10X,3112/10X,3112/                    NUD16340
     X          10X,3012/10X,3112/10X,3012/10X,3112 )                   NUD16350
        READ (LUNIN,502,END=6000) METPRT,NPPT,LUNPPT                    NUD16360
  502   FORMAT (10X,715)                                                NUD16370
        READ (LUNIN,503,END=6000) TSNOW                                 NUD16380
  503   FORMAT (10X,F10.0)                                              NUD16390
      METOPT = METPRT                                                   NUD16400
      WRITE (LUNPRT,705)                                                NUD16410
      WRITE (LUNPRT,706) (J,J=1,31)                                     NUD16420
      WRITE (LUNPRT,707) IDAYS                                          NUD16430
      WRITE (LUNPRT,701)                                                NUD16440
      IF (IABS (IDAYS(60)) .LE. 1) GO TO 7                              NUD16450
      DO 5 I1=61,366                                                    NUD16460
    5 IDAYS(I1-1) = IDAYS(I1)                                           NUD16470
      IDAYS(366) = 0                                                    NUD16480
    7 WRITE(LUNPRT,702) NPPT,(LUNPPT(L),L=1,NPPT)                       NUD16490
      WRITE(LUNPRT,708) TSNOW                                           NUD16500
      TSNOW = 273.15 + (TSNOW - 32.0) * 5.0/9.0                         NUD16510
      READ (LUNMET,END=10,ERR=10) IDSFC,IYRSFC,IDUPR,IYRUPR             NUD16520
      WRITE (LUNPRT,703) IDSFC,IYRSFC,IDUPR,IYRUPR                      NUD16530
C                                                                       NUD16540
      RETURN                                                            NUD16550
   10 WRITE (LUNPRT,704)                                                NUD16560
      RETURN                                                            NUD16570
  701 FORMAT(1H0,10X,'THE CODES IN THE ABOVE TABLE MEAN:'/              NUD16580
     X T15,'-1 = THIS DAY WILL BE SIMULATED USING MET. DATA FROM CARDS'/NUD16590
     X T15,' 0 = THIS DAY WILL NOT BE SIMULATED '/                      NUD16600
     X T15,'+1 = THIS DAY WILL BE SIMULATED USING PREPROCESSED MET. DATANUD16610
     X ON LOGICAL UNIT 8'/                                              NUD16620
     X 3X,'ANY OTHER VALUE ON FEBRUARY 29TH SIGNIFIES A NON-LEAP-YEAR CONUD16630
     XNDITION, AND THE PROGRAM ADJUSTS THE SEQUENCE ACCORDINGLY.'/)     NUD16640
  702  FORMAT(1H1,10X,'THE NUMBER OF INPUT PRECIPITATION FILES IS ',I2, NUD16650
     X     ',  UNIT NUMBERS ARE: ',5(I4,2X))                            NUD16660
  703  FORMAT(1H0,10X,'N. W. S. STATION IDENTIFIER FOR SURFACE MET. DATA:NUD16670
     X ',T71,I5/11X,'CALENDAR YEAR OF RECORD OF SURFACE MET. DATA:',T71,NUD16680
     X  3H 19,I2/                                                       NUD16690
     X  11X,'N. W. S. STATION IDENTIFIER FOR MIXING HEIGHT DATA:',      NUD16700
     X  T71,I5/11X,'CALENDAR YEAR OF RECORD OF MIXING HEIGHT DATA:',T71,NUD16710
     X  3H 19,I2/)                                                      NUD16720
  704  FORMAT(1H0,10X,'PREPROCESSED SURFACE METEOROLOGICAL DATA NOT FOUNDNUD16730
     X ON UNIT 8.'/11X,'PROGRAM CONTINUES ASSUMING HOURLY METEOROLOGICALNUD16740
     X DATA WILL BE INPUT FOR SELECTED DAYS '/11X,                      NUD16750
     X' ON UNIT 5 (STANDARD CARD INPUT UNIT)'/)                         NUD16760
  705  FORMAT(1H1,10X,'THE FOLLOWING TABLE INDICATES THOSE DAYS',       NUD16770
     X    ' WHICH WILL BE USED IN THIS RUN:'/)                          NUD16780
```

```
706   FORMAT (1X,'MONTH',20X,'DAYS'///T20,31I3//)                       NUD16790
707   FORMAT(1X,'JANUARY     ',T20,31I3/                                NUD16800
     X      1X,'FEBRUARY    ',T20,29I3/                                NUD16810
     X      1X,'MARCH       ',T20,31I3/                                NUD16820
     X      1X,'APRIL       ',T20,30I3/                                NUD16830
     X      1X,'MAY         ',T20,31I3/                                NUD16840
     X      1X,'JUNE        ',T20,30I3/                                NUD16850
     X      1X,'JULY        ',T20,31I3/                                NUD16860
     X      1X,'AUGUST      ',T20,31I3/                                NUD16870
     X      1X,'SEPTEMBER   ',T20,30I3/                                NUD16880
     X      1X,'OCTOBER     ',T20,31I3/                                NUD16890
     X      1X,'NOVEMBER    ',T20,30I3/                                NUD16900
     X      1X,'DECEMBER    ',T20,31I3/)                               NUD16910
708   FORMAT(1H0/11X,'THE THRESHOLD TEMPERATURE FOR FROZEN/LIQUID PRECIPNUD16920
     XITATION (DEGREES F) IS: ',F6.2,T100,'TSNOW')                      NUD16930
6000  STOP                                                              NUD16940
      END                                                               NUD16950
```

```
        SUBROUTINE KPOLLS                                               KP000010
        COMMON /ORIGIN/ XO,YO,UNIT,NEAST,NNORTH,NPOLLS                  KP000020
        COMMON /PTSRCS/ QXY(4,10),PARAMS(5,10), HTABL(3,10,10),         KP000030
       X                DISTBL(3,10,10),COORDS(2,10),IFLAG(10),KNTPTS   KP000040
        COMMON /ARSRCS/ VGRID,ACROSS,UP,QXYS(5,50),KNTARS               KP000050
        COMMON /CITIES/ QXYCIT(5,5),SIGZO(6,5),XTENT(6,5),IFLAGC(5),KNTCTYKP000060
        COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5) KP000070
C                                                                       KP000080
        DIMENSION QPOLLS(3,2)                                           KP000090
C                                                                       KP000100
1       DO 1000 KP = 1,2                                                KP000110
        IF( KNTPTS .EQ. 0) GO TO 101                                    KP000120
        DO 100 M1=1,KNTPTS                                              KP000130
100     QPOLLS(1,KP)=QPOLLS(1,KP) + QXY(KP,M1)                          KP000140
101     IF( KNTARS .EQ. 0) GO TO 201                                    KP000150
        DO 200 M2=1,KNTARS                                              KP000160
200     QPOLLS(2,KP)=QPOLLS(2,KP) + QXYS(KP,M2)                         KP000170
201     IF (KNTCTY .EQ. 0) GO TO 1000                                   KP000180
        DO 300 M3=1,KNTCTY                                              KP000190
300     QPOLLS(3,KP)=QPOLLS(3,KP) + QXYCIT(KP,M3)                       KP000200
1000    CONTINUE                                                        KP000210
C                                                                       KP000220
        NPOLLS=2                                                        KP000230
        WRITE (LUNPRT,600)                                              KP000240
C                                                                       KP000250
        DO 2000 KP = 1,2                                                KP000260
        TOTAL=QPOLLS(1,KP) + QPOLLS(2,KP) + QPOLLS(3,KP)                KP000270
        IF (TOTAL .GT. 0.0) GO TO 2000                                  KP000280
        NPOLLS = NPOLLS -1                                              KP000290
        WRITE(LUNPRT,601) KP                                            KP000300
2000    CONTINUE                                                        KP000310
C                                                                       KP000320
        WRITE (LUNPRT,602) NPOLLS,((QPOLLS(I,J),J=1,2),I=1,3)           KP000330
C                                                                       KP000340
        IF (NPOLLS .LE. 1)STOP                                          KP000350
        RETURN                                                          KP000360
600     FORMAT(1H1)                                                     KP000370
601     FORMAT(7X,'THERE ARE ZERO TOTAL EMISSIONS FROM ALL SOURCES ',   KP000380
       X  'FOR POLLUTANT ',I2,1H.)                                      KP000390
602     FORMAT(1H0,5X,I2,' POLLUTANTS WILL BE MODELLED IN THIS RUN.'/   KP000400
       X   10X,'TOTAL POINT SOURCE EMISSIONS:',T41,1P2G12.4/            KP000410
       X   10X,'TOTAL AREA SOURCE EMISSIONS:',T41,1P2G12.4/             KP000420
       X   10X,'TOTAL CITY SOURCE EMISSIONS:',T41,1P2G12.4/)            KP000430
        END                                                             KP000440
```

```
      SUBROUTINE GETMET(METPRT,IOK,IRON)                                NUD13940
      DIMENSION DDUMMY(24),HDUMMY(24)                                   NUD13950
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)  NUD13960
      COMMON /METDAT/IDAYS(366),THETA(24),U(24),KST(24),HL(24),TEMP(24),NUD13970
     X               IRAIN(25,5)                                        NUD13980
      COMMON /MCONST/ KNTDAY(12),NDAY,IYR,IMO,IDAYNO,L(5)               NUD13990
      IRON=0                                                            NUD14000
C                                                                       NUD14010
      NDAY=NDAY+1                                                       NUD14020
      IF(NDAY .GT. 366) GO TO 4000                                      NUD14030
      IF(IDAYS(NDAY)) 950,1001,1000                                     NUD14040
 1001 IOK=-1                                                            NUD14050
      RETURN                                                            NUD14060
  950 DO 960 K=1,24                                                     NUD14070
      READ (LUNIN,905,END=4000) IYR,IMO,IDAYNO,KST(K),U(K),TEMP(K),     NUD14080
     X     THETA(K),HL(K)                                               NUD14090
  960 CONTINUE                                                          NUD14100
  905 FORMAT(4I10,4F10.0)                                               NUD14110
      GO TO 2000                                                        NUD14120
 1000 CONTINUE                                                          NUD14130
      READ (LUNMET,ERR=5000,END=4000) IYR,IMO,DAYNO,KST,U,TEMP,         NUD14140
     X        DDUMMY,THETA,(HL(J),HDUMMY(J),J=1,24)                     NUD14150
      IDAYNO=DAYNO                                                      NUD14160
      IF (NDAY-IDAYNO) 4000,2000,1000                                   NUD14170
 2000 IF(NPPT .LE.0)GO TO 2025                                          NUD14180
      DO 2010 J=1,NPPT                                                  NUD14190
      DO 2001 K=1,25                                                    NUD14200
 2001 IRAIN(K,J)=0                                                      NUD14210
      LUNJ=LUNPPT(J)                                                    NUD14220
 2005 READ(LUNJ,712,ERR=5000,END=5000) JYR,JMO,JDAYNO,ICARD,            NUD14230
     X          (IRAIN(K,J),K=1,12)                                     NUD14240
      JDAYNO=JDAYNO+KNTDAY(JMO)                                         NUD14250
      IF(JMO .GT. 2 .AND. MOD(JYR+1900,4) .EQ. 0) JDAYNO=JDAYNO+1       NUD14260
      IF(IABS(IYR-JYR)+IABS(IMO-JMO)+IABS(IDAYNO-JDAYNO))5005,2004,2005 NUD14270
 2004 IF(ICARD .EQ. 9)GO TO 2010                                        NUD14280
      READ(LUNJ,713) (IRAIN(K,J),K=13,24)                               NUD14290
      DO 2002 K=1,24                                                    NUD14300
 2002 IRAIN(25,J)=IRAIN(25,J)+IRAIN(K,J)                                NUD14310
      IRON=IRON+IRAIN(25,J)                                             NUD14320
 2010 CONTINUE                                                          NUD14330
 2025 IF(METPRT .NE.0) WRITE(LUNPRT,710) IYR,IMO,IDAYNO                 NUD14340
      DO 1005 K=1,24                                                    NUD14350
      T= THETA(K)-180.                                                  NUD14360
      IF(T .LT. 0.0) T=T+360.                                           NUD14370
      IF(T .GT. 360.) T=T-360.                                          NUD14380
      THETA(K)=T                                                        NUD14390
      ISTAB=KST(K)                                                      NUD14400
      IF(ISTAB .GT. 6)  KST(K)=6                                        NUD14410
      IF (METPRT .NE. 0)WRITE (LUNPRT,711) K,T,U(K),KST(K),HL(K),TEMP(K)NUD14420
      IF(NPPT .EQ. 0 .OR. METPRT .EQ. 0)GO TO 1005                      NUD14430
      WRITE(LUNPRT,716) (IRAIN(K,J),J=1,NPPT)                           NUD14440
 1005 CONTINUE                                                          NUD14450
      IF(METPRT .EQ. 0) RETURN                                          NUD14460
      IF(NPPT.GT.0) WRITE(LUNPRT,717)(IRAIN(25,J),J=1,NPPT)             NUD14470
      RETURN                                                            NUD14480
 5000 STOP 42                                                           NUD14490
 5005 WRITE(LUNPRT,602) IYR,JYR,IMO,JMO,IDAYNO,JDAYNO,LUNJ              NUD14500
```

```
      STOP 52                                                           NUD14510
 4000 WRITE(LUNPRT,601)                                                 NUD14520
      IOK=1                                                             NUD14530
      RETURN                                                            NUD14540
  601 FORMAT(1H1,'   END OF INPUT HOURLY MET. DATA ENCOUNTERED.',       NUD14550
     X    'NORMAL END OF RUN.'/)                                        NUD14560
  602 FORMAT(1H1,'     BAD DATE MATCH:  YEARS=',2I4,',  MONTHS=',2I4,   NUD14570
     X     ',  DAYNUMBERS=',2I5/11X,'PRECIP. DATA INPUT UNIT NO.:',I3/) NUD14580
  710 FORMAT(1H1,20X,'DAILY METEOROLOGICAL DATA FOR YEAR:',I3,' MONTH:',NUD14590
     X  I3,'  DAY NO.:',I4/                                             NUD14600
     X    T14,'HOUR',T20,'WIND DIR',T30,'WIND SPD',T40,'P-G STAB',T50,  NUD14610
     X     'MIX.HGT',T60,'AIR TEMP',T72,'RAINFALL AMOUNTS'/)            NUD14620
  711 FORMAT(12X,I3,T17,F10.3,T27,F10.3,T42,I2,T47,F10.3,T57,F10.3)     NUD14630
  712 FORMAT(1X,3I2,I1,12I6)                                            NUD14640
  713 FORMAT(8X,12I6)                                                   NUD14650
  716 FORMAT(1H+,70X,5I8)                                               NUD14660
  717 FORMAT(12X,'DAILY RAINFALL TOTAL:',T65,5I8)                       NUD14670
      END                                                               NUD14680
      SUBROUTINE PRESET
      COMMON /DISKOP/ DAYDEP(2,225),PPTCON(2,20,24),DPLCON(2,20,24)
      COMMON /VALUES/ ADJWD(2),TAU(2),
     X                UDEPOS(2),USETTL(2),SCAVNG(2),BKGRND(2),RUFLEN
      COMMON /ORIGIN/ XO,YO,UNIT,NEAST,NNORTH,NPOLLS
      COMMON /RECEPS/ INDEX(225),INDPPT(20),KATCH(20),
     X                NEXAMS(10),KNTEX
C
C=== **************************************************************
      COMMON / TEST / IDBUG,LUNDBG
      LOGICAL IDBUG
C=== **************************************************************
C
C     IT IS A KEY ASSUMPTION THAT 'BACKGROUND' CONCENTRATIONS, AS USED
C     IN THIS  APPROACH, ARE INVARIANT WITH AVERAGING TIME AND LOCATION.
C
      NRECEP = NNORTH * NEAST
      DO 26 LR = 1,NRECEP
      DO 26 M = 1,NPOLLS
26    DAYDEP(M,LR) = 0.0
C
      DO 24 KHR = 1,24
      CALL HOURLY (T,TRAD,W,TA,HM,ISTAB,KHR)
      DO 23 M = 1,NPOLLS
      DO 25   LK=1,20
      IF (KATCH(LK) .EQ. 0)  GO TO 25
      PPTCON(M,LK,KHR) = BKGRND(M)
      DPLCON(M,LK,KHR) = BKGRND(M)
25    CONTINUE
      BKGFLX = FLUX (M, BKGRND(M))
21    DO 22 LR = 1,NRECEP
22    DAYDEP(M,LR) = DAYDEP(M,LR) + BKGFLX * 3600.
23    CONTINUE
24    CONTINUE
C
100   IF (.NOT. IDBUG) RETURN
C=== **************************************************************
C===  PUT DEBUG PRINTS HERE.....
      DO 125 K1 = 1,2
      WRITE (LUNDBG,800) K1,(K3,K3=1,24)
      DO 124 K2=1,20
      IF (KATCH(K2) .EQ. 0) GO TO 124
      WRITE (LUNDBG,801) K2,(PPTCON(K1,K2,K3),K3=1,24),
     X                   K2,(DPLCON(K1,K2,K3),K3=1,24)
124   CONTINUE
      WRITE (LUNDBG,802) K1,(DAYDEP(K1,K2),K2=1,NRECEP)
125   CONTINUE
800   FORMAT (1H1,3X,'*** PRESET TABLES:  ****  POLLUTANT:',I3,
     X             ' ON THE CURRENT DAY  *****' //
     X         2X,'KATCH:',2X,12I10/10X,12I10)
801   FORMAT (4X,I3,3X,1P12G10.2/10X,12G10.2/)
802   FORMAT (2X,I3,' DAYDEP BY REC.:',1P12G9.2/(26X,12G9.2/))
C=== **************************************************************
C
300   RETURN
      END
      SUBROUTINE HOURLY (T,TRAD,WSPD,TA,HMIX,ISTAB,KHR)
      COMMON /ORIGIN/ XO,YO,UNIT,NEAST,NNORTH,NPOLLS
      COMMON /VALUES/ ADJWD(2),TAU(2),
     X                VDEPOS(2),VSETTL(2),SCAVNG(2),BKGRND(2),RUFLEN
      COMMON /METDAT/ IDY(366),THETA(24),U(24),KST(24),HL(24),TEMP(24),
     X                IRAIN(25,5),TSNOW
C
C=== ***********************************************************
      COMMON / TEST / IDBUG,LUNDBG
      LOGICAL IDBUG
C=== ***********************************************************
C
      ISTAB=KST(KHR)
      T=THETA(KHR)
      TRAD=T * 0.0174533
      WSPD=U(KHR)
      TA = TEMP (KHR)
      HMIX= HL(KHR)
C
      DO 20 M=1,NPOLLS
C
20    ADJWD(M)=VD(VDEPOS(M),VSETTL(M),ISTAB,WSPD,RUFLEN)
C
100   IF (.NOT. IDBUG) RETURN
C=== ***********************************************************
C===  PUT DEBUG PRINTS HERE.....
C=== ***********************************************************
C
      RETURN
      END
      SUBROUTINE DUMMY                                                  NUD13700
C                                                                       T1 00010
C=== *************************************************************     T1 00020
      COMMON / TEST / IDBUG,LUNDBG                                      T1 00030
      LOGICAL IDBUG                                                     T1 00040
C=== *************************************************************     T1 00050
C                                                                       T1 00060
C                                                                       T2 00010
100   IF (.NOT. IDBUG) RETURN                                           T2 00020
C=== *************************************************************     T2 00030
C===  PUT DEBUG PRINTS HERE.....                                        T2 00040
C=== *************************************************************     T2 00050
C                                                                       T2 00060
      RETURN                                                            NUD13710
      END                                                               NUD13720
      SUBROUTINE XYLOC (X,Y,J,NCOL,NROW)                                NUD24700
      COMMON /ORIGIN/   XSWCOR,YSWCOR,UNIT,NEAST,NNORTH                 NUD24710
      NCOL=((J-1)/NNORTH)+1                                             NUD24720
      NROW=J-(NCOL-1)*NNORTH                                            NUD24730
      X = (NCOL-0.5)*UNIT+XSWCOR                                        NUD24740
      Y = (NROW-0.5)*UNIT+YSWCOR                                        NUD24750
      RETURN                                                            NUD24760
      END                                                               NUD24770
      SUBROUTINE PTGEOM(RECX,RECY,KREC,KHR,ISTAB,IKST,IWIND,HMIX,W,TRAD)NUD19130
      COMMON /RECEPS/ INDEX(225),INDPPT(20),KATCH(20),                  NUD19140
     X                NEXAMS(10),KNTEX                                  NUD19150
      COMMON /PTSRCS/ QXY(4,10),PARAMS(5,10),HTABL(3,10,10),            NUD19160
     X                DISTBL(3,10,10),COORDS(2,10),IFLAG(10),KNTPTS     NUD19170
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)  NUD19180
      COMMON /METDAT/ IDAYS(366),THETA(24),U(24),KST(24),HL(24),        NUD19190
     X                TEMP(24),IRAIN(25,5)                              NUD19200
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,                       NUD19210
     X                T,P,YCOEFF(4),YPOWER(4),ZCOEFF(4),ZPOWER(4),      NUD19220
     X                R2PI,ROOT2,ROOTPI,                                NUD19230
     X                J1STAB(7),J2STAB(7)                               NUD19240
C                                                                       NUD19250
C=== ***************************************************************    NUD19260
      COMMON / TEST / IDBUG,LUNDBG                                      NUD19270
      LOGICAL IDBUG                                                     NUD19280
C=== ***************************************************************    NUD19290
C                                                                       NUD19300
      IKST=J1STAB(ISTAB)                                                NUD19310
      SINT=SIN(TRAD)                                                    NUD19320
      COST=COS(TRAD)                                                    NUD19330
C     SELECT SOURCE DATA                                                NUD19340
      DO 2500 KSRC=1,KNTPTS                                             NUD19350
      X=0.                                                              NUD19360
      Y=0.                                                              NUD19370
      SRCX=QXY(3,KSRC)                                                  NUD19380
      SRCY=QXY(4,KSRC)                                                  NUD19390
C                                                                       NUD19400
      R=SRCX-RECX                                                       NUD19410
      S=SRCY-RECY                                                       NUD19420
      X=S*COST+R*SINT                                                   NUD19430
      IF(X.GT.0.1)GO TO 2020                                            NUD19440
 2001 IFLAG(KSRC)=-1                                                    NUD19450
C    IFLAG= -1 MEANS RECEPTOR IS UPWIND OR TO CLOSE TO SOURCE           NUD19460
C    (X .LE. 100.0 METERS ) OR                                          NUD19470
C    RECEPTOR IS 'TOO FAR' OFF-AXIS (ARCTAN(Y/X) .GT. PLUME WIDTH)      NUD19480
      GO TO 2400                                                        NUD19490
 2020 Y=S*SINT-R*COST                                                   NUD19500
      PHI=ATAN(Y/X)                                                     NUD19510
      IF(ABS(PHI)-WIDTH(ISTAB))2022,2022,2001                           NUD19520
 2022 CONTINUE                                                          NUD19530
C      DETERMINE IF RECEPTOR IS BEYOND DISTANCE-TO-FINAL-PLUME-RISE     NUD19540
C      FOR NOMINAL STABILITY AND WIND SPEED FOR THIS HOUR               NUD19550
C      FIRST DETERMINE CLOSEST TABULATED WIND SPEED VALUE               NUD19560
      DELMIN=50.                                                        NUD19570
      DO 2023 L=1,10                                                    NUD19580
      DELU=ABS(W-WSPDS(L))                                              NUD19590
      IF(DELU .GT. DELMIN)GO TO 2025                                    NUD19600
      DELMIN=DELU                                                       NUD19610
 2023 CONTINUE                                                          NUD19620
      L=11                                                              NUD19630
 2025 IWIND=L-1                                                         NUD19640
      IF(X - DISTBL(IKST,IWIND,KSRC))2030,2050,2050                     NUD19650
 2030 IFLAG(KSRC)=1                                                     NUD19660
C    IFLAG= 1 MEANS RECEPTOR IS CLOSER THAN FINAL-RISE DISTANCE         NUD19670
C    IFLAG= 0 MEANS RECEPTOR IS FURTHER THAN FINAL-RISE DISTANCE        NUD19680
      GO TO 2400                                                        NUD19690
```

```
2050  IFLAG(KSRC)=0                                                          NUD19700
2400  COORDS(1,KSRC)=X                                                       NUD19710
      COORDS(2,KSRC)=Y                                                       NUD19720
2500  CONTINUE                                                               NUD19730
C                                                                            NUD19740
100   IF (.NOT. IDBUG) RETURN                                                NUD19750
C=== *************************************************************           NUD19760
C===  PUT DEBUG PRINTS HERE.....                                             NUD19770
C=== *************************************************************           NUD19780
C                                                                            NUD19790
      RETURN                                                                 NUD19800
      END                                                                    NUD19810
      SUBROUTINE PTCONC(WSPD,TA,HM,IST,IKST,IWIND,KATCH,KREC,KHR)
      COMMON /ORIGIN/ X0,Y0,UNIT,NEAST,NNORTH,NPOLLS
      COMMON /PTSRCS/ QXY(4,10),PARAMS(5,10),HTABL(3,10,10),
     X                DISTBL(3,10,10),COORDS(2,10),IFLAG(10),KNTPTS
      COMMON /VALUES/ ADJWD(2),TAU(2),
     X                UDEPOS(2),USETTL(2),SCAVNG(2),BKGRND(2),RUFLEN
      COMMON /MCONST/ KNTDAY(12),NDAY,IYEAR,IMONTH,IDAYNO,L(5)
      COMMON /CHITAB/ HRMAXP(2,5),HRMAXA(2,5),HRMAXC(2,5),
     X                MAXHRP(2,5,2),MAXHRA(2,5,2),MAXHRC(2,5,2),
     X                DAYMAX(2,10),MAXDAY(2,10)
      COMMON /DISKOP/ DAYDEP(2,225),PPTCON(2,20,24)
C
C=== ***********************************************************
      COMMON / TEST / IDBUG,LUNDBG
      LOGICAL IDBUG
C=== ***********************************************************
C
C
      KST=IST-1
      IF (KST .LT. 1) KST=1
      IF (KST .GT. 5) KST=5
      JST=KST
      IF(JST .GT. 4) JST=4
C
      DO 2100 KSRC=1,KNTPTS
      IF(IFLAG(KSRC)) 2100,2050,2030
2030  HP=PARAMS(1,KSRC)
      VF=PARAMS(5,KSRC)
      TS=PARAMS(4,KSRC)
      F=3.12139*VF*(TS-TA)/TS
      XKM=COORDS(1,KSRC)
      CALL PLRISE(HFINAL,HEFF,DISTF,IST,WSPD,XKM,DTHDZ,TA,P,HP,F)
      GO TO 2055
2050  HEFF=HTABL(IKST,IWIND,KSRC)
      XKM=COORDS(1,KSRC)
2055  CALL HEFFIX(HEFF,KSRC,IST,WSPD)
      YKM=COORDS(2,KSRC)
      TTIME=XKM/(3.6*WSPD)
C       TTIME (TRAVEL OR TRANSPORT TIME) BY THIS EQN. WILL BE IN HOURS
C         SINCE WIND SPEED HAS BEEN CONVERTED FROM M/SEC TO KM/HOUR
C         AND X  SHOULD BE IN KM ALREADY
      DO 1000 M=1,NPOLLS
      Q=QXY(M,KSRC)
      IF (Q .EQ. 0.0) GO TO 1000
      DECAY=0.0
      PTCON=0.0
      PTDEP=0.0
      HLIVES=TTIME/TAU(M)
C   FOR DIMENSIONAL CONSISTENCY, TAU SHOULD ALSO BE IN UNITS OF HOURS
      IF (HLIVES-50.0) 10,1000,1000
10    DECAY=EXP(- 0.6931472 * HLIVES)
      CHI = CONC (Q,WSPD,XKM,YKM,HEFF,HM,JST,ADJWD(M),USETTL(M),VTHICK)
      PTCON=1.0E6*CHI*DECAY
C
      CALL UPDATE (M,1,KREC,KSRC,KATCH,KHR,PTCON,TTIME,VTHICK)
C
1000  CONTINUE
```

```
C                                                                       PTC00580
100   IF (.NOT. IDBUG) GO  TO 2100                                      PTC00590
C=== **************************************************************     PTC00600
C===  PUT DEBUG PRINTS HERE.....                                        PTC00610
C=== **************************************************************     PTC00620
C                                                                       PTC00630
2100  CONTINUE                                                          PTC00640
C                                                                       PTC00650
      RETURN                                                            PTC00660
      END                                                               PTC00670
      SUBROUTINE ARGEOM                                                 ARG00010
      COMMON /ARANGL/ EPS(24),IEPS(24),ISIGNX(24),ISIGNY(24),ITABX(24), ARG00020
     1               ITABY(24),INCRX(24),INCRY(24)                      ARG00030
      COMMON /METDAT/ IDAYS(366),THETA(24),U(24),KST(24),HL(24),        ARG00040
     X        TEMP(24),IRAIN(25,5)                                      ARG00050
      COMMON /ACONST/ EPSLIM(10),AAR(6),BAR(6),ITABLE(5,10)             ARG00060
C                                                                       ARG00070
C                                                                       ARG00080
C=== ************************************************************       ARG00090
      COMMON / TEST / IDBUG,LUNDBG                                      ARG00100
      LOGICAL IDBUG                                                     ARG00110
C=== ************************************************************       ARG00120
C                                                                       ARG00130
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ARG00140
      DO 2423 KHR=1,24                                                  ARG00150
      WVEC= THETA(KHR)/57.29578+ 3.14159265                             ARG00160
      IF(WVEC.GT.6.2831853) WVEC=WVEC - 6.2831853                       ARG00170
C                                                                       ARG00180
C                                                                       ARG00190
C     IWV HELPS DETERMINE THE PATTERN OF GRID SQUARES AFFECTED BY EACH  ARG00200
C     AREA SOURCE.  PATTERN DEPENDS ONLY ON WIND DIRECTION.             ARG00210
C     EPS(KHR) IS THE ANGLE BETWEEN THE WIND VECTOR AND THE NEAREST     ARG00220
C     COORDINATE AXIS.                                                  ARG00230
C                                                                       ARG00240
      IWV= 1 + WVEC/0.78540                                             ARG00250
      GO TO (2411,2412,2413,2414,2415,2416,2417,2418),IWV               ARG00260
 2411 ITABX(KHR)=1                                                      ARG00270
      ISIGNX(KHR)=1                                                     ARG00280
      ISIGNY(KHR)=1                                                     ARG00290
      EPS(KHR)= WVEC                                                    ARG00300
      GO TO 2419                                                        ARG00310
 2412 ITABX(KHR)=0                                                      ARG00320
      ISIGNX(KHR)=1                                                     ARG00330
      ISIGNY(KHR)=1                                                     ARG00340
      EPS(KHR)= 1.5708 - WVEC                                           ARG00350
      GO TO 2419                                                        ARG00360
 2413 ITABX(KHR)=0                                                      ARG00370
      ISIGNX(KHR)=1                                                     ARG00380
      ISIGNY(KHR)=-1                                                    ARG00390
      EPS(KHR)= WVEC - 1.57080                                          ARG00400
      GO TO 2419                                                        ARG00410
 2414 ITABX(KHR)=1                                                      ARG00420
      ISIGNX(KHR)=1                                                     ARG00430
      ISIGNY(KHR)=-1                                                    ARG00440
      EPS(KHR)= 3.1416 - WVEC                                           ARG00450
      GO TO 2419                                                        ARG00460
 2415 ITABX(KHR)=1                                                      ARG00470
      ISIGNX(KHR)=-1                                                    ARG00480
      ISIGNY(KHR)=-1                                                    ARG00490
      EPS(KHR)= WVEC - 3.1416                                           ARG00500
      GO TO 2419                                                        ARG00510
 2416 ITABX(KHR)=0                                                      ARG00520
      ISIGNX(KHR)=-1                                                    ARG00530
      ISIGNY(KHR)=-1                                                    ARG00540
      EPS(KHR)= 4.7124 - WVEC                                           ARG00550
      GO TO 2419                                                        ARG00560
 2417 ITABX(KHR)=0                                                      ARG00570
```

```
      ISIGNX(KHR)=-1                                                    ARG00580
      ISIGNY(KHR)=1                                                     ARG00590
      EPS(KHR)= WVEC - 4.7124                                           ARG00600
      GO TO 2419                                                        ARG00610
 2418 ITABX(KHR)=1                                                      ARG00620
      ISIGNX(KHR)=-1                                                    ARG00630
      ISIGNY(KHR)=1                                                     ARG00640
      EPS(KHR)= 6.2832 - WVEC                                           ARG00650
 2419 IF(ITABX(KHR).EQ.0) GO TO 2420                                    ARG00660
      ITABY(KHR)=0                                                      ARG00670
      INCRX(KHR)=0                                                      ARG00680
      INCRY(KHR)=1                                                      ARG00690
      GO TO 2421                                                        ARG00700
 2420 ITABY(KHR)=1                                                      ARG00710
      INCRX(KHR)=1                                                      ARG00720
      INCRY(KHR)=0                                                      ARG00730
 2421 DO 2422 L=1,10                                                    ARG00740
      IF(EPS(KHR).GT.EPSLIM(L)) GO TO 2422                              ARG00750
      IEPS(KHR)=L                                                       ARG00760
      GO TO 2423                                                        ARG00770
 2422 CONTINUE                                                          ARG00780
      IEPS(KHR)=10                                                      ARG00790
 2423 CONTINUE                                                          ARG00800
      RETURN                                                            ARG00810
      END                                                               ARG00820
      SUBROUTINE ARCONC (KHR,KSTAB,WS)
      COMMON /ORIGIN/ XSWC,YSWC,GRID,LX,LY,NPOLLS
      COMMON /ACONST/ EPSLIM(10),AAR(6),BAR(6),ITABLE(5,10)
      COMMON /ARANGL/ TEPS(24),KEPS(24),ISIGNX(24),ISIGNY(24),
     X                ITABX(24),ITABY(24),INCRX(24),INCRY(24)
      COMMON /ARSRCS/ VGRID,ELX,ELY,QXYS(5,50),KNTARS
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,
     X                T,P,YCOEFF(4),YPOWER(4),ZCOEFF(4),ZPOWER(4),
     X                R2PI,ROOT2,ROOTPI,
     X                J1STAB(7),J2STAB(7)
C
C=== ******************************************************************
      COMMON / TEST / IDBUG,LUNDBG
      LOGICAL IDBUG
C=== ******************************************************************
C
      DIMENSION QN(5),ERATE(2),JJ(3)
C
C
C  CALCULATE COEFFICIENTS FOR GIFFORD-HANNA AREA SOURCE ALGORITHM.
C
2423  IA= KSTAB-1
      IF(IA.EQ.0) IA=1
      AA= AAR(IA)
      BA= 1. - BAR(IA)
      IF (IA .GT. 5) IA=5
      EPS=TEPS(KHR)
      QC= (0.79788456*(500.*GRID/COS(EPS))**BA)/(WS*AA*BA)
      Q3= 3.**BA
      Q5= 5.**BA
      Q7= 7.**BA
      Q9= 9.**BA
      QN(1)= 1.
      QN(2)= Q3 - 1.
      QN(3)= Q5 - Q3
      QN(4)= Q7 - Q5
      QN(5)= Q9 - Q7
C
C  LOOP THROUGH ALL AREA SOURCES.  EMISSIONS FROM EACH AREA SOURCE
C    ARE APPORTIONED AMONG THE RECEPTOR GRID SQUARES WHOLLY OR
C    PARTIALLY COVERED BY THE SOURCE.  EACH AFFECTED RECEPTOR GRID
C    SQUARE IS TREATED AS A SOURCE.
C
      DO 2429 K=1,KNTARS
      ESUM=0.0
      DO 25 IJK=1,NPOLLS
      QX=QXYS(IJK,K)
      ERATE(IJK)=QX
25    ESUM=ESUM + QX
      IF (ESUM.EQ.0.0) GO TO 2429
      SIZE=QXYS(5,K)
      AREA= SIZE*SIZE
      TIMMAX = 5.0*ROOT2*SIZE / (3.6*WS)
      AX1= QXYS(3,K)
      AX2= AX1 + SIZE
      NX1= (AX1-XSWC)*VGRID + 5.
      NX2= (AX2-XSWC)*VGRID + 5.
      IF(NX1.GT.LX+8.OR.NX2.LT.1) GO TO 2429                            ARC00580
      IF(NX1.GE.1) GO TO 32                                             ARC00590
      NX1= 1                                                            ARC00600
      AX1= XSWC - 4.*GRID                                               ARC00610
   32 IF(NX2.LE.LX+8) GO TO 34                                          ARC00620
      NX2= LX+8                                                         ARC00630
      AX2= XSWC + (ELX + 4.)*GRID                                       ARC00640
   34 X1= XSWC + (FLOAT(NX1)-5.)*GRID                                   ARC00650
      X2= XSWC + (FLOAT(NX2)-4.)*GRID                                   ARC00660
      AY1= QXYS(4,K)                                                    ARC00670
      AY2= AY1 + SIZE                                                   ARC00680
      NY1= (AY1-YSWC)*VGRID + 5.                                        ARC00690
      NY2= (AY2-YSWC)*VGRID + 5.                                        ARC00700
      IF(NY1.GT.LY+8.OR.NY2.LT.1) GO TO 2429                            ARC00710
      IF(NY1.GE.1) GO TO 36                                             ARC00720
      NY1= 1                                                            ARC00730
      AY1= YSWC - 4.*GRID                                               ARC00740
   36 IF(NY2.LE.LY+8) GO TO 38                                          ARC00750
      NY2= LY+8                                                         ARC00760
      AY2= YSWC + (ELY + 4.)*GRID                                       ARC00770
   38 Y1= YSWC + (FLOAT(NY1)-5.)*GRID                                   ARC00780
      Y2= YSWC + (FLOAT(NY2)-4.)*GRID                                   ARC00790
      DO 2428 IX=NX1,NX2                                                ARC00800
      IF(NX1.NE.NX2) GO TO 10                                           ARC00810
      DX= AX2-AX1                                                       ARC00820
      GO TO 13                                                          ARC00830
   10 IF(IX.NE.NX1) GO TO 11                                            ARC00840
      DX= GRID+X1-AX1                                                   ARC00850
      GO TO 13                                                          ARC00860
   11 IF(IX.NE.NX2) GO TO 12                                            ARC00870
      DX= GRID+AX2-X2                                                   ARC00880
      GO TO 13                                                          ARC00890
   12 DX=GRID                                                           ARC00900
   13 DO 2427 JY=NY1,NY2                                                ARC00910
      IF(NY1.NE.NY2) GO TO 20                                           ARC00920
      DY= AY2-AY1                                                       ARC00930
      GO TO 23                                                          ARC00940
   20 IF(JY.NE.NY1) GO TO 21                                            ARC00950
      DY= GRID+Y1-AY1                                                   ARC00960
      GO TO 23                                                          ARC00970
   21 IF(JY.NE.NY2) GO TO 22                                            ARC00980
      DY= GRID+AY2-Y2                                                   ARC00990
      GO TO 23                                                          ARC01000
   22 DY=GRID                                                           ARC01010
   23 SPLIT= DX*DY/(AREA*GRID*GRID)                                     ARC01020
      CALL ARCOMP (KHR,K,IX,JY,IA,QN,ERATE,QC,SPLIT,TIMMAX)             ARC01030
C                                                                       ARC01040
C=== *************************************************************** ARC01050
C=== PUT DEBUG PRINTS HERE.....                                         ARC01060
      IF (IDBUG) WRITE (LUNDBG,999) KHR,K,IA,AA,BA,QC,QN,               ARC01070
     X            TIMMAX,SPLIT,DX,DY                                    ARC01080
  999 FORMAT (2X,'* ARCONC:  KHR,K,IA =',3I5,                           ARC01090
     X       3X,'TIMMAX,SPLIT,DX,DY =',3X,4F6.3/                        ARC01100
     X       3X,'AA,BA,QC=',1X,3F6.2,2X,'QN(1-5)=',1X,5F6.3)            ARC01110
C=== *************************************************************** ARC01120
C                                                                       ARC01130
 2427 CONTINUE                                                          ARC01140
 2428  CONTINUE                                                           ARC01150
 2429  CONTINUE                                                           ARC01160
       RETURN                                                             ARC01170
       END                                                                ARC01180
```

```
      SUBROUTINE CTGEOM(TRAD,ISTAB)                                     NUD08920
      COMMON /ORIGIN/ XO,YO,UNIT,NEAST,NNORTH,NPOLLS                    NUD08930
      COMMON /CITIES/ QXYCIT(5,5),SIGZO(6,5),XTENT(6,5),IFLAGC(5),KNTCTYNUD08940
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,                       NUD08950
     X                T,PRESS,YCOEFF(4),YPOWER(4),ZCOEFF(4),ZPOWER(4),  NUD08960
     X                R2PI,ROOT2,ROOTPI,                                NUD08970
     X                J1STAB(7),J2STAB(7)                               NUD08980
C                                                                       NUD08990
C=== ***************************************************************    NUD09000
      COMMON / TEST / IDBUG,LUNDBG                                      NUD09010
      LOGICAL IDBUG                                                     NUD09020
C=== ***************************************************************    NUD09030
C                                                                       NUD09040
      COMPLEX CTCENT,P(4)                                               NUD09050
C                                                                       NUD09060
      SINT=SIN(TRAD)                                                    NUD09070
      COST=COS(TRAD)                                                    NUD09080
      IASME = J2STAB(ISTAB)                                             NUD09090
      PWR = 1.0/YPOWER (IASME)                                          NUD09100
      FACT = YCOEFF (IASME)                                             NUD09110
      DO 5000 I1=1,KNTCTY                                               NUD09120
      B=0.5 * QXYCIT(5,I1) * (ABS(SINT) + ABS(COST) )                   NUD09130
      DWMAX=-9999.                                                      NUD09140
      DWMIN= 9999.                                                      NUD09150
      CWMAX=-9999.                                                      NUD09160
      CWMIN= 9999.                                                      NUD09170
      IFLAGC(I1) = -9                                                   NUD09180
      XTEST = 1.0E-3 * ((1.0E3 * B)/(6.5 * FACT)) * * PWR               NUD09190
1     CTCENT = CMPLX(QXYCIT(3,I1), QXYCIT(4,I1) )                       NUD09200
      P(1) = CTCENT -CMPLX(XO,YO)                                       NUD09210
      P(2) = CTCENT - CMPLX (XO, YO + FLOAT(NNORTH) * UNIT)             NUD09220
      P(3) = CTCENT - CMPLX (XO + FLOAT(NEAST) * UNIT,                  NUD09230
     X                       YO + FLOAT(NNORTH) * UNIT)                 NUD09240
      P(4) = CTCENT - CMPLX (XO + FLOAT(NEAST) * UNIT, YO )             NUD09250
      DO 4950 I2 =1,4                                                   NUD09260
      DOWNW = SINT * REAL(P(I2)) + COST * AIMAG(P(I2))                  NUD09270
      CROSSW= SINT * AIMAG(P(I2)) - COST * REAL(P(I2))                  NUD09280
      DWMAX = AMAX1(DWMAX,DOWNW)                                        NUD09290
      DWMIN = AMIN1(DWMIN,DOWNW)                                        NUD09300
      CWMAX = AMAX1(CWMAX,ABS(CROSSW))                                  NUD09310
      CWMIN = AMIN1(CWMIN,ABS(CROSSW))                                  NUD09320
4950  CONTINUE                                                          NUD09330
      XTENT(1,I1) = DWMIN                                               NUD09340
      XTENT(2,I1) = DWMAX                                               NUD09350
      XTENT(3,I1) = CWMIN                                               NUD09360
      XTENT(4,I1) = CWMAX                                               NUD09370
      XTENT(5,I1) = B                                                   NUD09380
      XTENT(6,I1) = XTEST                                               NUD09390
C     DETERMINE IF RECEPTOR GRID IS DOWNWIND AND BEYOND THE CITY: IF NOTNUD09400
C     THEN IFLAGC = -1                                                  NUD09410
      IF (DWMIN .GT. B) GO TO 20                                        NUD09420
      IFLAGC(I1) = -1                                                   NUD09430
      GO TO 5000                                                        NUD09440
C     NEXT CHECK WHETHER THE RECEPTOR GRID IS IN NEAR-FIELD OR FAR-FIELDNUD09450
C     OF THE SOURCE:  CRITERION IS WHETHER DWMAX .LE. XTEST, WHERE XTESTNUD09460
C     IS THE DOWNWIND DISTANCE SUCH THAT 2.15 * SIGMA-Y(XTEST,IASME)=B/3NUD09470
C     OR (APPROXIMATELY) SIGMA-Y =(1/6.5) * B                           NUD09480
```

```
 20    IF (DWMAX .LT. XTEST) GO TO 1000
       IFLAGC(I1) = 50
       XVIRTY = 1.0E-3 * (( 1.0E3 * B) / (FACT) ) * *  PWR
       XTENT(1,I1) = XVIRTY
       GO TO 5000
 1000  CONTINUE
 C     NEAR-FIELD CONDITIONS VALID FOR DOWNWIND EXTENT: NOW EXAMINE
 C     CROSSWIND EXTENT
       IF (CWMAX .GT. B) GO TO 21
 C     THE RECEP GRID IS COMPLETELY UNDER THE PLUME IN THE NEAR-FIELD
       IFLAGC(I1) = 0
       GO TO 5000
 21    IF (CWMIN .LT.2.0* B) GO TO 22
       IFLAGC(I1) = -1
 C     THE RECEP GRID IS TOO FAR OFF-AXIS (MORE THAN 2*B) IN THE NR FIELD
       GO TO 5000
 22    IFLAGC(I1) = 1
 C     IFLAGC=1 REPRESENTS ANY SORT OF 'MIXED' IMPACTS: GRID IS TOO WIDE/
 C     TOO LONG TO MODEL WITH COMPLETE NEGLECT OF LATERAL DISPERSION
 5000  CONTINUE
 C
 100   IF (.NOT. IDBUG) RETURN
 C=== ******************************************************************
 C===  PUT DEBUG PRINTS HERE.....
 C=== ******************************************************************
 C
       RETURN
       END
```

```
      SUBROUTINE CTCONC(HL,TRAD,WSPD,XR,YR,KR,KATCH,KC,KHR,KST,IFLAG)   CTC00010
      COMMON /CITIES/ QXYCIT(5,5),SIGZ0(6,5),XTENT(6,5),IFLAGC(5),KNTCTYCTC00020
      COMMON /ORIGIN/ X0,Y0,UNIT,NEAST,NNORTH,NPOLLS                    CTC00030
      COMMON /VALUES/ ADJWD(2),TAU(2),                                  CTC00040
     X                UDEPOS(2),USETTL(2),SCAVNG(2),BKGRND(2),RUFLEN    CTC00050
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,                       CTC00060
     X                T,P,YCOEFF(4),YPOWER(4),ZCOEFF(4),ZPOWER(4),      CTC00070
     X                R2PI,ROOT2,ROOTPI,                                CTC00080
     X                J1STAB(7),J2STAB(7)                               CTC00090
C                                                                       CTC00100
C=== ***********************************************************         CTC00110
      COMMON / TEST / IDBUG,LUNDBG                                      CTC00120
      LOGICAL IDBUG                                                     CTC00130
C=== ***********************************************************         CTC00140
C                                                                       CTC00150
      IASME = J2STAB(KST)                                               CTC00160
      DELX = QXYCIT(3,KC) - XR                                          CTC00170
      DELY = QXYCIT(4,KC) - YR                                          CTC00180
      DOWNW = SIN(TRAD) * DELX + COS(TRAD) * DELY - XTENT(5,KC)         CTC00190
      CROSSW = SIN(TRAD) * DELY - COS(TRAD) * DELX                      CTC00200
      CRATIO = ABS(CROSSW)/XTENT(5,KC)                                  CTC00210
      IF(IFLAG .EQ. 1 .AND. CRATIO .GT. 2.0) GO TO 2000                 CTC00220
      TTIME = DOWNW/(3.6 * WSPD)                                        CTC00230
      AREA = 1.0E6 * QXYCIT(5,KC) * * 2                                 CTC00240
      DWM = 1.0E3 * DOWNW                                               CTC00250
      CWM = 1.0E3 * CROSSW                                              CTC00260
      SY = SYCOMP (DWM,IASME)                                           CTC00270
      SZ = SIGZ0(KST,KC) + SZCOMP(DWM,IASME)                            CTC00280
      XVIRTZ = (SZ/ZCOEFF(IASME)) * * (1.0/ZPOWER(IASME))               CTC00290
      VTHICK = SZ                                                       CTC00300
C                                                                       CTC00310
      DO 1000 M= 1,NPOLLS                                               CTC00320
      Q = QXYCIT(M,KC)                                                  CTC00330
      QBAR = Q/AREA                                                     CTC00340
      IF (QBAR .EQ. 0.0) GO TO 1000                                     CTC00350
      DECAY = 0.0                                                       CTC00360
      CTCON = 0.0                                                       CTC00370
      HLIVES = TTIME/TAU(M)                                             CTC00380
      IF (HLIVES - 50.0) 10,1000,1000                                   CTC00390
10    DECAY = EXP (-0.6931472 * HLIVES)                                 CTC00400
      IF (IFLAG .EQ. 50) GO TO 50                                       CTC00410
      T1 = 1.0 - ZPOWER(IASME)                                          CTC00420
      CHI = 2.0*(QBAR * XVIRTZ **T1) / (R2PI*WSPD*ZCOEFF(IASME)*T1)     CTC00430
      IF (IFLAG .EQ. 0) GO TO 100                                       CTC00440
      IF (CRATIO .LE. 1.0) GO TO 100                                    CTC00450
      OFFAX = 1.0E3 * (CRATIO -1.0)                                     CTC00460
      CHI = CHI * EXCOMP (OFFAX/SY)                                     CTC00470
      GO TO 100                                                         CTC00480
50    XVIRTY = 1.0E3 *(DOWNW + XTENT(5,KC) + XTENT(1,KC) )              CTC00490
      SYVIRT = SYCOMP (XVIRTY,IASME)                                    CTC00500
CC***   APPLY CONCENTRATION/DEPOSITION EQUATIONS -  GENERALLY CONSISTENCTC00510
CC***   TREATMENT WITH 'REAL' POINT SOURCES                             CTC00520
CC    DOWNWIND EXTENT OF NO MIXING LID INFLUENCE IN THE Z DIMENSION     CTC00530
CC    IS GIVEN BY XLCOMP...RECALL DOWNW = 0.0 IS AT D/W EDGE OF CITY, NOCTC00540
CC    NOT THE CENTER                                                    CTC00550
      XL = XLCOMP (HL,IASME)                                            CTC00560
      XRATC = XVIRTZ / XL                                               CTC00570
      XRATD = DWM / XL                                                  CTC00580
   55 IF(IASME .NE. 4 .AND. HL .LT. 2000.0 .AND. XRATD .GT. 1.0)GO TO 80CTC00590
C                                                                       CTC00600
C EMULATE ERMAK...                                                      CTC00610
C                                                                       CTC00620
   60 DIFFZ = WSPD * SZ * SZ / (2.0 * XVIRTZ)                           CTC00630
      CHI = (Q/WSPD) * CERMAK (XVIRTY,SYVIRT,0.0,SZ,DIFFZ,              CTC00640
     X       ADJWD(M),USETTL(M) )                                       CTC00650
      VTHICK=SZ                                                         CTC00660
      GO TO 100                                                         CTC00670
   80 CONTINUE                                                          CTC00680
C     BEYOND XL, WITH RESPECT TO DISPERSION FROM THE VIRTUAL POINT SRCE CTC00690
C     BUT, MASS DEPLETION OCCURS ONLY FROM THE DOWNWIND EDGE OF THE CITYCTC00700
C     *************************                                         CTC00710
      UPW = XVIRTZ - DWM                                                CTC00720
      XA = 2.0 * XL - UPW                                               CTC00730
      IF (XA .GT. 0.0) GO TO 82                                         CTC00740
      XA = 2.0 * DWM                                                    CTC00750
      WRITE (LUNDBG,91) XL,XVIRTZ,DWM                                   CTC00760
   91 FORMAT('  PROBLEM IN CTCONC W/ XA VARIABLE: (XL,XVIRTZ,DWM) =',   CTC00770
     X       1P3G12.4)                                                  CTC00780
   82 SZXA = SZCOMP(XA,IASME)                                           CTC00790
      DIFFXA = WSPD*SZXA**2/(2.0*XA)                                    CTC00800
   51 RATE=DEPRAT( 0.0,SZXA,DIFFXA,ADJWD(M),USETTL(M) )                 CTC00810
      DFACT=1.0-AMAX1(RATE,0.0)                                         CTC00820
      DFACT=AMAX1(DFACT,0.0)                                            CTC00830
      VTHICK=SZXA                                                       CTC00840
C                                                                       CTC00850
      IF(DWM.GE. XA ) GO TO 95                                          CTC00860
C                                                                       CTC00870
   81 SYXA = SYCOMP(XA,IASME)                                           CTC00880
      YXA = CWM * (XA/DWM)                                              CTC00890
C                                                                       CTC00900
   83 CXA =Q*DFACT*EXCOMP(YXA/SYXA)/(R2PI*WSPD*SYXA*HL)                 CTC00910
C                                                                       CTC00920
C                                                                       CTC00930
      XB = XL - UPW                                                     CTC00940
      IF ( XB .GE. 0.0 ) GO TO 85                                       CTC00950
      XB = DWM                                                          CTC00960
      WRITE (LUNDBG,92) XL,UPW,XVIRTZ,DWM                               CTC00970
   92 FORMAT(' SECOND TROUBLE IN CTCONC: XB .LE. 0.0. (XL,UPW,XVIRTZ,DWMCTC00980
     X)=',1P4G12.4)                                                     CTC00990
   85 SYXB=SYCOMP(XB,IASME)                                             CTC01000
      SZXB=SZCOMP(XB,IASME)                                             CTC01010
      DIFFXB=WSPD*SZXB**2/(2.0*XB)                                      CTC01020
      YXB=YXA * 0.5                                                     CTC01030
   87 RELC=CERMAK(YXB,SYXB, 0.0 ,SZXB,DIFFXB,ADJWD(M),USETTL(M) )       CTC01040
      CXB=(Q/WSPD)*RELC                                                 CTC01050
C                                                                       CTC01060
      XFACT=ALOG( DWM/XB )* 1.442695                                    CTC01070
      CXBLOG=-99.9                                                      CTC01080
      IF(CXB.GT.0.0) CXBLOG=ALOG(CXB)                                   CTC01090
      CXALOG=-99.9                                                      CTC01100
      IF (CXA .GT. 0.0 ) CXALOG = ALOG( CXA )                           CTC01110
   88 CONLOG=CXBLOG*(1.0-XFACT)+CXALOG*XFACT                            CTC01120
      CHI=EXP(CONLOG)                                                   CTC01130
   89 SZXLOG=ALOG(SZXB)*(1.0-XFACT)+ALOG(SZXA)*XFACT                    CTC01140
```

```
      VTHICK=EXP(SZXLOG)                                                CTC01150
C     PERFORM LOG-LOG INTERPOLATION                                     CTC01160
      GO TO 100                                                         CTC01170
95    CONTINUE                                                          CTC01180
C*****     SOURCE DEPLETION MODEL     *****                             CTC01190
C     FOR DISTANCES BEYOND 2*XL (AS MODIFIED),ASSUME UNIFORM CONC IN Z  CTC01200
      DFACT2=EXP(-ADJWD(M) * (DWM-XA)/(WSPD*HL))                        CTC01210
C     COMPUTE DEPLETION FACTOR                                          CTC01220
96    CHI=Q*DFACT*DFACT2*EXCOMP( CWM /SY)/(R2PI*WSPD*SY*HL)             CTC01230
      VTHICK=HL                                                         CTC01240
      GO TO 100                                                         CTC01250
C     **************************                                        CTC01260
100   CTCON = 1.0E 6 * CHI * DECAY                                      CTC01270
      CALL UPDATE (M,3,KR,KC,KATCH,KHR,CTCON,TTIME,VTHICK)              CTC01280
1000  CONTINUE                                                          CTC01290
C                                                                       CTC01300
2000  IF (.NOT. IDBUG) RETURN                                           CTC01310
C=== *************************************************************      CTC01320
C===  PUT DEBUG PRINTS HERE.....                                        CTC01330
      WRITE (LUNDBG,999) IASME,KHR,IFLAG,DOWNW,CROSSW,CRATIO,OFFAX,TTIMECTC01340
999   FORMAT (2X,'* CTCONC(1): * IASME,KHR,IFLAG =',3I3,2X,             CTC01350
     X    'DNW,CRW,CRATIO,OFFAX=',4F7.3,2X,'TT=',F7.3)                  CTC01360
      WRITE (LUNDBG,998) SY,SZ,XVIRTZ,XVIRTY,SYVIRT,VTHICK,CHI,XL       CTC01370
998   FORMAT (2X,'* CTCONC(2): * SY,SZ,XV-Z,XV-Y=',4F9.2,               CTC01380
     X      2X,'SYV,VTH,CHI,XL=',4F9.2)                                 CTC01390
      WRITE (LUNDBG,997) XRATC,XRATD,UPW,XFACT,                         CTC01400
     X                   XA,SZXA,YXA,XB,SZXB,YXB                        CTC01410
997   FORMAT (2X,'* CTCONC(3): * XR-C,XR-D,UPW,XF=',                    CTC01420
     X 4F9.2/9X,'XA,SZXA,YXA,XB,SZXB,YXB=',6F9.2)                       CTC01430
C=== *************************************************************      CTC01440
C                                                                       CTC01450
      RETURN                                                            CTC01460
      END                                                               CTC01470
      SUBROUTINE DAYEND(IRON)                                           DAY00010
      COMMON /ORIGIN/ X0,Y0,UNIT,NEAST,NNORTH,NPOLLS                    DAY00020
      COMMON /RECEPS/ INDEX(225),INDPPT(20),KATCH(20),                  DAY00030
     X                NEXAMS(10),KNTEX                                  DAY00040
      COMMON /DISKOP/ DAYDEP(2,225),PPTCON(2,20,24),DPLCON(2,20,24)     DAY00050
      COMMON /VALUES/ ADJWD(2),TAU(2),                                  DAY00060
     X                UDEPOS(2),USETTL(2),SCAVNG(2),BKGRND(2),RUFLEN    DAY00070
      COMMON /ANSTAT/ ANNAVG(2,20,2),KDAYS                              DAY00080
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)  DAY00090
      COMMON /CHITAB/ HRMAXP(2,5),HRMAXA(2,5),HRMAXC(2,5),              DAY00100
     X                MAXHRP(2,5,2),MAXHRA(2,5,2),MAXHRC(2,5,2),        DAY00110
     X                DAYMAX(2,10), MAXDAY(2,10)                        DAY00120
C                                                                       DAY00130
C=== ***************************************************************    DAY00140
      COMMON / TEST / IDBUG,LUNDBG                                      DAY00150
      LOGICAL IDBUG                                                     DAY00160
C=== ***************************************************************    DAY00170
C                                                                       DAY00180
      DIMENSION AVFLUX(2,20),SUMCON(2,20),J(3)                          DAY00190
C                                                                       DAY00200
      NR=NNORTH * NEAST                                                 DAY00210
      KDAYS = KDAYS + 1                                                 DAY00220
C                                                                       DAY00230
      DO 9999 M=1,NPOLLS                                                DAY00240
      DO 9999 K=1,20                                                    DAY00250
      SUMCON(M,K)=0.0                                                   DAY00260
      AVFLUX(M,K)=0.0                                                   DAY00270
9999  CONTINUE                                                          DAY00280
C                                                                       DAY00290
      DO 500 KR = 1,NR                                                  DAY00300
      CALL INDXER (INDEX(KR),IDUM,J(1),J(2),J(3))                       DAY00310
      DO 400 ID1 = 1,IDUM                                               DAY00320
      ID2 = J (ID1)                                                     DAY00330
      DO 300 M = 1,NPOLLS                                               DAY00340
300   AVFLUX(M,ID2) = AVFLUX(M,ID2) + DAYDEP(M,KR)                      DAY00350
C                                                                       DAY00360
400   CONTINUE                                                          DAY00370
C                                                                       DAY00380
500   CONTINUE                                                          DAY00390
C                                                                       DAY00400
      DO 1001 KATID=1,20                                                DAY00410
      KAREA=KATCH(KATID)                                                DAY00420
      IF(KAREA .EQ. 0) GO TO 1001                                       DAY00430
      AREA = FLOAT(KAREA)                                               DAY00440
      DO 200 KHR = 1,24                                                 DAY00450
      DO 150 M = 1,NPOLLS                                               DAY00460
150   SUMCON (M,KATID) = SUMCON (M,KATID) + PPTCON (M,KATID,KHR)        DAY00470
C                                                                       DAY00480
200   CONTINUE                                                          DAY00490
C                                                                       DAY00500
      DO 999 M = 1,NPOLLS                                               DAY00510
C                                                                       DAY00520
      CHI = SUMCON(M,KATID)/(24.0*AREA)                                 DAY00530
      CALL DAYUPD(M,KATID,CHI)                                          DAY00540
      SUMCON(M,KATID) = CHI                                             DAY00550
      ANNAVG(M,KATID,1) = ANNAVG(M,KATID,1) + CHI                       DAY00560
      ANNAVG(M,KATID,2) = ANNAVG(M,KATID,2) + ALOG(CHI)                 DAY00570
```

```
C                                                                       DAY00580
      AVFLUX(M,KATID) = AVFLUX(M,KATID)/AREA                            DAY00590
999   CONTINUE                                                          DAY00600
1001  CONTINUE                                                          DAY00610
C                                                                       DAY00620
C                                                                       DAY00630
100   IF (.NOT. IDBUG) GO TO 1002                                       DAY00640
C=== ************************************************************        DAY00650
C===  PUT DEBUG PRINTS HERE.....                                        DAY00660
      DO 125 K1 = 1,2                                                   DAY00670
      WRITE (LUNDBG,800) K1,KDAYS,(K3,K3=1,24)                          DAY00680
      DO 124 K2=1,20                                                    DAY00690
      IF (KATCH(K2) .EQ. 0) GO TO 124                                   DAY00700
      WRITE (LUNDBG,801) K2,(PPTCON(K1,K2,K3),K3=1,24),                 DAY00710
     X                   K2,(DPLCON(K1,K2,K3),K3=1,24)                  DAY00720
      WRITE (LUNDBG,802) K2,AVFLUX(K1,K2),SUMCON(K1,K2),                DAY00730
     X                            (ANNAVG(K1,K2,K3),K3=1,2)             DAY00740
124   CONTINUE                                                          DAY00750
125   CONTINUE                                                          DAY00760
800   FORMAT (1H1,3X,'*** DAYEND TABLES:  ****  POLLUTANT:',I3,         DAY00770
     X              ' ON THE ',I3,'-TH DAY  *****' //                   DAY00780
     X        2X,'KATCH:',2X,12I10/10X,12I10)                           DAY00790
801   FORMAT (4X,I3,3X,1P12G10.2/10X,12G10.2/)                          DAY00800
802   FORMAT (4X,I3,' AVFLUX + SUM: ',1P2G12.2,                         DAY00810
     X        7X,' ANNARIAV + ANNGEOAV: ',2G12.2/)                      DAY00820
C=== ************************************************************        DAY00830
C                                                                       DAY00840
C                                                                       DAY00850
1002  CALL OUTPUT(AVFLUX,IRON)                                          DAY00860
C                                                                       DAY00870
      RETURN                                                            DAY00880
      END                                                               DAY00890
      SUBROUTINE WRAPUP (NPOLLS,KNTPTS,KNTARS,KNTCTY)
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)
      COMMON /CHITAB/ HRMAXP(2,5),HRMAXA(2,5),HRMAXC(2,5),
     X                MAXHRP(2,5,2),MAXHRA(2,5,2),MAXHRC(2,5,2),
     X                DAYMAX(2,10), MAXDAY(2,10)
      COMMON /RECEPS/ INDEX(225),INDPPT(20),KATCH(20),
     X                NEXAMS(10),KNTEX
      COMMON /ANSTAT/ ANNAVG(2,20,2),KDAYS
C
C=== ************************************************************
      COMMON / TEST / IDBUG,LUNDBG
      LOGICAL IDBUG
C=== ************************************************************
C
7000  CONTINUE
      DAYS = FLOAT (KDAYS)
      DO 2000 M=1,NPOLLS
      WRITE (LUNPRT,700)
      WRITE (LUNPRT,800) M,KDAYS
      IF(KNTPTS .EQ. 0) GO TO 2
1     WRITE (LUNPRT,701) M,(N,HRMAXP(M,N),MAXHRP(M,N,1),MAXHRP(M,N,2),
     X          N=1,5)
2     IF(KNTARS .EQ. 0) GO TO 3
      WRITE (LUNPRT,702) M,(N,HRMAXA(M,N),MAXHRA(M,N,1),MAXHRA(M,N,2),
     X          N=1,5)
3     IF(KNTCTY .EQ. 0) GO TO 4
      WRITE (LUNPRT,703) M,(N,HRMAXC(M,N),MAXHRC(M,N,1),MAXHRC(M,N,2),
     X          N=1,5)
4     WRITE (LUNPRT,704) M,(N,DAYMAX(M,N),MAXDAY(M,N),N=1,10)
C
      DO 1000 K= 1,20
      IF (KATCH(K) .EQ. 0) GO TO 1000
      ARIAVG = ANNAVG(M,K,1)/DAYS
      GEOAVG = EXP (ANNAVG(M,K,2)/DAYS)
      WRITE (LUNPRT,801) K,ARIAVG,GEOAVG
1000  CONTINUE
C
2000  CONTINUE
C
100   IF (.NOT. IDBUG) RETURN
C=== ************************************************************
C===  PUT DEBUG PRINTS HERE.....
C=== ************************************************************
C
      RETURN
C
700   FORMAT (1H1)
701   FORMAT(10X,'HIGHEST POINT SOURCE HOURLY CONCENTRATION RANKING FOR
     X POLLUTANT:',I2/5(15X,' RANK:',I2,' CONCENTRATION:',1PG12.4,
     X    '  (RECEP,SOURCE,HOUR,DAY):',2I8/)/)
702   FORMAT(10X,'HIGHEST AREA  SOURCE HOURLY CONCENTRATION RANKING FOR
     X POLLUTANT:',I2/5(15X,' RANK:',I2,' CONCENTRATION:',1PG12.4,
     X    '  (RECEP,SOURCE,HOUR,DAY):',2I8/)/)
703   FORMAT(10X,'HIGHEST CITY  SOURCE HOURLY CONCENTRATION RANKING FOR
     X POLLUTANT:',I2/5(15X,' RANK:',I2,' CONCENTRATION:',1PG12.4,
     X    '  (RECEP,SOURCE,HOUR,DAY):',2I8/)/)
704   FORMAT(10X,'TEN HIGHEST DAILY AVERAGE CONCENTRATION RANKING FOR ',
     X'  POLLUTANT:',I2/ 10(15X,' RANK:',I2,' CONCENTRATION:',1PG12.4,  WRA00580
     X    '  (KATCH,DAY):',I8/)/)                                       WRA00590
C                                                                       WRA00600
801   FORMAT (5X,'PERIOD-OF-RECORD AVERAGES FOR CATCHMENT ',I2,         WRA00610
     X 1H:,2X,1PG12.4,' (ARITHMETIC MEAN);',G12.4,' (GEOMETRIC MEAN).') WRA00620
800   FORMAT (10X,'***** STATISTICAL TABLES FOR POLLUTANT ',I1,         WRA00630
     X   ', PERIOD-OF-RECORD: ',I3,' DAYS. *****'//)                    WRA00640
C                                                                       WRA00650
      END                                                               WRA00660
      SUBROUTINE INDXCK (J,N,J1,J2,J3,IFLAG)                            NUD15150
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)  NUD15160
      IFLAG = 0                                                         NUD15170
      N=0                                                               NUD15180
      J1=0                                                              NUD15190
      J2=0                                                              NUD15200
      J3=0                                                              NUD15210
      IF (J .GE. 1E7)  STOP  10                                         NUD15220
      IF (J .LT. 1E4) GO TO 3                                           NUD15230
      N=3                                                               NUD15240
      J1=J/10000                                                        NUD15250
      J2= (J-10000*J1)/100                                              NUD15260
      J3= (J-10000*J1-100*J2)                                           NUD15270
      IF(J2 .NE. 0 .AND. J3 .NE. 0)GO TO 20                             NUD15280
      WRITE (LUNPRT,601)                                                NUD15290
      IFLAG = -1                                                        NUD15300
      RETURN                                                            NUD15310
   20 IF(J1-J2 .NE. 0 .AND. J2-J3 .NE. 0 .AND. J1-J3 .NE. 0) RETURN     NUD15320
      WRITE (LUNPRT,602)                                                NUD15330
  601 FORMAT(2X,'*** ONE CATCHMENT INDEX IS EQUAL TO ZERO...')          NUD15340
  602 FORMAT(2X,'*** TWO CATCHMENT INDEXES EQUAL EACH OTHER...')        NUD15350
      IFLAG = -1                                                        NUD15360
      RETURN                                                            NUD15370
    3 IF (J .LT. 1E2) GO TO 4                                           NUD15380
      N=2                                                               NUD15390
      J1=J/100                                                          NUD15400
      J2=J-100*J1                                                       NUD15410
      IF (J2 .NE. 0) GO TO 40                                           NUD15420
      WRITE (LUNPRT,601)                                                NUD15430
      IFLAG = -1                                                        NUD15440
      RETURN                                                            NUD15450
   40  IF (J1-J2 .NE. 0) RETURN                                         NUD15460
      WRITE (LUNPRT,602)                                                NUD15470
      IFLAG = -1                                                        NUD15480
      RETURN                                                            NUD15490
    4 N=1                                                               NUD15500
      J1=J                                                              NUD15510
      IF (J1 .NE. 0) GO TO 50                                           NUD15520
      WRITE (LUNPRT,601)                                                NUD15530
      IFLAG = -1                                                        NUD15540
   50 RETURN                                                            NUD15550
      END                                                               NUD15560
      SUBROUTINE INDXER (J,N,J1,J2,J3)
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)  NUD15570
      N=0                                                               NUD15580
      J1=0                                                              NUD15590
      J2=0                                                              NUD15600
      J3=0                                                              NUD15610
      IF (J .LT. 1E4) GO TO 3                                           NUD15620
      N=3                                                               NUD15630
      J1=J/10000                                                        NUD15640
      J2= (J-10000*J1)/100                                              NUD15650
      J3= (J-10000*J1-100*J2)                                           NUD15660
      RETURN                                                            NUD15670
    3 IF (J .LT. 1E2) GO TO 4                                           NUD15680
      N=2                                                               NUD15690
      J1=J/100                                                          NUD15700
      J2=J-100*J1                                                       NUD15710
      RETURN                                                            NUD15720
    4 N=1                                                               NUD15730
      J1=J                                                              NUD15740
      RETURN                                                            NUD15750
      END                                                               NUD15760
                                                                        NUD15770
      FUNCTION FLUX( M, CONC )                                          NUD13880
      COMMON /VALUES/ ADJWD(2),TAU(2),                                  NUD13890
     X             UDEPOS(2),USETTL(2),SCAVNG(2),BKGRND(2),RUFLEN       NUD13900
      FLUX= CONC * ADJWD (M)                                            NUD13910
      RETURN                                                            NUD13920
      END                                                               NUD13930
      FUNCTION VD (WD,WS,ISTAB,U,Z0)                                    NUD23260
C                                                                       NUD23270
C      JST IS A 5-CLASS STABILITY PARAMETER MAPPED FROM THE P-G 7-CLASS NUD23280
C     SCHEME GIVEN BY THE CALLING PGM TO THE 5-CLASS SCHEME USED HERE   NUD23290
C                                                                       NUD23300
C                                                                       NUD23310
      COMMON /MCONST/ KNTDAY(12),NDAY,IYR,IMO,IDAYNO,L(5)               NUD23320
C          L MAPS JST TO MONIN-OBUKHOV LENGTH (METERS)                  NUD23330
C                                                                       NUD23340
C                                                                       NUD23350
      JST=ISTAB-1                                                       NUD23360
      IF (JST .LT. 1) JST=1                                             NUD23370
      IF (JST .GT. 5) JST=5                                             NUD23380
C                                                                       NUD23390
C          LOW DEPOSITION VELOCITY OVERRIDE (LIQUID PHASE CONTROL):     NUD23400
C                                                                       NUD23410
C          ASSUMED UNITS ARE METERS/SEC.                                NUD23420
C                                                                       NUD23430
      IF (WD .GT. 0.001) GO TO 5                                        NUD23440
      VD=WD                                                             NUD23450
      RETURN                                                            NUD23460
C                                                                       NUD23470
C          THE DEFINITION OF FRICTION VELOCITY DEPENDENCE ON STABILITY  NUD23480
C              (REFERENCE:  BUSINGER, ET. AL.,  J. APPL. METEOROLOGY, 1971)NUD23490
C                ASSUMING REFERENCE HEIGHT OF 10 METERS                 NUD23500
C                                                                       NUD23510
5     LMO=L (JST)                                                       NUD23520
      XLMO = FLOAT(LMO)                                                 NUD23530
      IF (83. - XLMO) 10,20,20                                          NUD23540
10    USTAR = 0.4 * U / (2.3026 - ALOG(Z0) + 30./XLMO +                 NUD23550
     X             510./(XLMO * XLMO))                                  NUD23560
      GO TO 30                                                          NUD23570
20    USTAR = 0.4 * U /(2.3026 - ALOG(Z0) + 47./XLMO)                   NUD23580
C                                                                       NUD23590
C          DEFINITION OF PHASE-TRANSFER COEFFICIENT                     NUD23600
C                                                                       NUD23610
30    WA = (USTAR * USTAR / U) / (1. + 8. * USTAR / U)                  NUD23620
      VD = WA + WS                                                      NUD23630
      RETURN                                                            NUD23640
      END                                                               NUD23650
      SUBROUTINE PLRISE (HF,HX,DISTF,KST,U,X,DTHDZ,T,P,HP,F)            NUD17710
      IF(P .LE. 0.0)P=980.                                              NUD17720
      GO TO (7,7,7,7,22,23,23),KST                                      NUD17730
7     IF(F-55.)8,9,9                                                    NUD17740
8     XST=14.*F**.625                                                   NUD17750
      GO TO 10                                                          NUD17760
9     XST=34.*F**.4                                                     NUD17770
10    DISTF=3.5*XST                                                     NUD17780
      DELHF=1.6*F**0.333333*DISTF**0.666667/U                           NUD17790
32    XM = 1000.* X                                                     NUD17800
14    DELHX = 1.6*F**0.333333*XM**0.666667/U                            NUD17810
      IF(DELHX.GT.DELHF)DELHX=DELHF                                     NUD17820
      GO TO 30                                                          NUD17830
   22 DTHDZ = 0.02                                                      NUD17840
      GO TO 24                                                          NUD17850
   23 DTHDZ = 0.035                                                     NUD17860
   24 S = 9.80616*DTHDZ/T                                               NUD17870
      DHA = 2.6*(F/(U*S))**0.333333                                     NUD17880
      DELHF = 5.0*F**0.25/S**0.375                                      NUD17890
      IF(DHA-DELHF) 25,25,27                                            NUD17900
   25 DELHF = DHA                                                       NUD17910
   27 DISTF = 3.14159*U/S**0.5                                          NUD17920
   33 XM = 1000.*X                                                      NUD17930
      IF(XM-DISTF)14,14,28                                              NUD17940
   28 DELHX = DELHF                                                     NUD17950
      GO TO 30                                                          NUD17960
   30 HX = HP + DELHX                                                   NUD17970
   31 HF = HP + DELHF                                                   NUD17980
      DISTF = DISTF/1000.                                               NUD17990
      RETURN                                                            NUD18000
      END                                                               NUD18010
      SUBROUTINE HEFFIX (HEFF,KSRC,ISTAB,U)                              NUD14690
      COMMON /PTSRCS/ QXY(4,10),PARAMS(5,10),HTABL(3,10,10),             NUD14700
     X                DISTBL(3,10,10),COORDS(2,10),IFLAG(10),KNTPTS      NUD14710
 7000 CONTINUE                                                           NUD14720
      C=0.0                                                              NUD14730
      STKVEL=PARAMS(3,KSRC)                                              NUD14740
      IF(STKVEL .GE. 1.5*U) GO TO 10                                     NUD14750
      C=3.0*PARAMS(2,KSRC)*(1.5-STKVEL/U)                                NUD14760
   10 IF( C .EQ. 0.0)RETURN                                              NUD14770
      HEFF1 = HEFF- C                                                    NUD14780
      IF (HEFF1 - 0.5*HEFF) 11,11,15                                     NUD14790
   11 HEFF = 0.5*HEFF                                                    NUD14800
      RETURN                                                             NUD14810
   15 HEFF=HEFF1                                                         NUD14820
      RETURN                                                             NUD14830
      END                                                                NUD14840
      FUNCTION CONC (Q,U,XKM,YKM,HEFF,HLID,KST,WD,WS,SZX)               CON00010
C     ROUTINE TO EVALUATE DOWNWIND CONCENTRATION                        CON00020
C     PARAMETERS:                                                       CON00030
C     Q--SOURCE EMISSION RATE (G/S)                                     CON00040
C     U--WIND SPEED (M/S)                                               CON00050
C     XKM--DOWNWIND DISTANCE (KM)                                       CON00060
C     YKM--CROSSWIND DISTANCE (KM)                                      CON00070
C     HEFF--EFFECTIVE SOURCE HEIGHT (M)                                 CON00080
C     HLID--MIXING HEIGHT (M)                                           CON00090
C     KST--STABILITY CLASS (4 LEVEL ASME)                               CON00100
C     WD--DEPOSITION VELOCITY (M/S)                                     CON00110
C     WS--GRAVITATIONAL SETTLING VELOCITY (M/S)                         CON00120
C     SZX--EFFECTIVE PLUME THICKNESS WITHIN HLID                        CON00130
C     CONC--POLLUTANT CONCENTRATION (G/M**3)                            CON00140
C                                                                       CON00150
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,                       CON00160
     X                T,P,YCOEFF(4),YPOWER(4),ZCOEFF(4),ZPOWER(4),      CON00170
     X                R2PI,ROOT2,ROOTPI,                                CON00180
     X                J1STAB(7),J2STAB(7)                               CON00190
      REAL KZ,KZXL,KZ2XL                                                CON00200
C                                                                       CON00210
C===  ******************************************************************CON00220
      COMMON / TEST / IDBUG,LUNDBG                                      CON00230
      LOGICAL IDBUG                                                     CON00240
C===  ******************************************************************CON00250
C                                                                       CON00260
      IF(HEFF.LE.HLID.AND.XKM.GT.0.1) GO TO 10                          CON00270
      CONC=0.0                                                          CON00280
      SZX=0.0                                                           CON00290
      GO TO 100                                                         CON00300
10    CONTINUE                                                          CON00310
      X=XKM*1000.0                                                      CON00320
      Y=YKM*1000.0                                                      CON00330
      SY=SYCOMP(X,KST)                                                  CON00340
      XL=XLCOMP(HLID,KST)                                               CON00350
C     COMPUTE NEARFIELD BOUNDARY:  SZ(XL) = 0.47*HLID                   CON00360
      IF(KST.NE.4.AND.HLID.LT.2000.0.AND.X.GT.XL) GO TO 50              CON00370
C     USE ERMAK SOLUTION FOR CONCENTRATION IN FOLLOWING CASES:          CON00380
C        1)  STABLE CONDITIONS                                          CON00390
C        2)  'INFINITE' MIXING HEIGHT                                   CON00400
C        3)  NEARFIELD REGION                                           CON00410
      SZ=SZCOMP(X,KST)                                                  CON00420
      KZ=U*SZ**2/(2.0*X)                                                CON00430
C     ESTIMATE VERTICAL DIFFUSION COEFFICIENT                           CON00440
11    RELC=CERMAK(Y,SY,HEFF,SZ,KZ,WD,WS)                                CON00450
      CONC=(Q/U)*RELC                                                   CON00460
      SZX=SZ                                                            CON00470
C     GET ERMAK CONCENTRATION                                           CON00480
      GO TO 100                                                         CON00490
50    CONTINUE                                                          CON00500
C*****     NON-ERMAK REGIME    *****                                    CON00510
      SZ2XL=SZCOMP(2.0*XL,KST)                                          CON00520
      KZ2XL=U*SZ2XL**2/(4.0*XL)                                         CON00530
51    RATE=DEPRAT(HEFF,SZ2XL,KZ2XL,WD,WS)                               CON00540
C     COMPUTE NET DEPOSITION RATE FROM 0 TO 2.0*XL                      CON00550
      DFACT=1.0-AMAX1(RATE,0.0)                                         CON00560
      DFACT=AMAX1(DFACT,0.0)                                            CON00570
```

```
C     COMPUTE SOURCE DEPLETION FACTOR AT 2*XL
      IF(X.GE.2.0*XL) GO TO 75
C*****      XL LT X LT 2*XL     *****
52    SY2XL=SYCOMP(2.0*XL,KST)
      Y2XL=Y*(2.0*XL/X)
C     COMPUTE CROSSWIND DISTANCE AT 2*XL
53    C2XL=Q*DFACT*EXCOMP(Y2XL/SY2XL)/(R2PI*U*SY2XL*HLID)
C     COMPUTE CONCENTRATION AT 2*XL USING SOURCE DEPLETION MODEL
C     CONCENTRATION AT 2*XL ASSUMED UNIFORM IN Z
      SYXL=SYCOMP(XL,KST)
      SZXL=SZCOMP(XL,KST)
      KZXL=U*SZXL**2/(2.0*XL)
      YXL=Y2XL*0.5
54    RELC=CERMAK(YXL,SYXL,HEFF,SZXL,KZXL,WD,WS)
      CXL=(Q/U)*RELC
C     COMPUTE CONCENTRATION AT DISTANCE XL
      XFACT=ALOG(X/XL)*1.442695
      CXLLOG=-99.9
      IF(CXL.GT.0.0) CXLLOG=ALOG(CXL)
      C2XLLG=-99.9
      IF(C2XL.GT.0.0) C2XLLG=ALOG(C2XL)
73    CONLOG=CXLLOG*(1.0-XFACT)+C2XLLG*XFACT
74    CONC=EXP(CONLOG)
      SZXLOG=ALOG(SZXL)*(1.0-XFACT)+ALOG(SZ2XL)*XFACT
      SZX=EXP(SZXLOG)
C     PERFORM LOG-LOG INTERPOLATION BETWEEN XL AND 2*XL
      GO TO 100
75    CONTINUE
C*****      SOURCE DEPLETION MODEL      *****
C     FOR DISTANCES BEYOND 2*XL, ASSUME UNIFORM CONC IN Z
      DFACT2=EXP(-WD*(X-2.0*XL)/(U*HLID))
C     COMPUTE DEPLETION FACTOR FROM 2*XL TO X
      SY=SYCOMP(X,KST)
76    CONC=Q*DFACT*DFACT2*EXCOMP(Y/SY)/(R2PI*U*SY*HLID)
      SZX=HLID
C
100   IF (.NOT. IDBUG) RETURN
C=== *************************************************************
C===  PUT DEBUG PRINTS HERE.....
      FCNVAL = CONC
      WRITE (LUNDBG,999) FCNVAL,U,HLID,XKM,YKM,XL,SZX,SZXL,SZ2XL,
     X       SY,RATE,XFACT,WD,WS,KST
999   FORMAT (' * CONC: * FCNVAL,U,HLID,XKM,YKM =',
     X 1P5G12.4/4X,'XL,SZX,SZXL,SZ2XL,SY =',5G9.2/4X,'RATE,XFACT,',
     X 'WS=',4G9.2,3X,'KST=',I2)
C=== *************************************************************
C
      RETURN
      END
      SUBROUTINE UPDATE (M,KTYPE,KREC,KSRC,KATCH,KHR,CONC,TTIME,VTHICK)
      COMMON /DISKOP/ DAYDEP(2,225),PPTCON(2,20,24),DPLCON(2,20,24)
      COMMON /CHITAB/ HRMAX(2,5,3),
     X                MAXHR(2,5,2,3),
     X                DAYMAX(2,10),MAXDAY(2,10)
      COMMON /MCONST/ KNTDAY(12),NDAY,IYEAR,IMONTH,IDAYNO,L(5)
C
C=== ***********************************************************
      COMMON / TEST / IDBUG,LUNDBG
      LOGICAL IDBUG
C=== ***********************************************************
C
      DIMENSION JJ(3)
C
C       COMPARE CONC TO THE CURRENT 5 HIGHEST HOURLY SOURCE
C       CONTRIBUTIONS FOR THIS SOURCE TYPE AT ANY RECEPTOR FOR THIS HOUR
C
      IF (CONC .LT. HRMAX (M,5,KTYPE)) GO TO 500
C
C       THIS VALUE OF CONC  BELONGS IN THE TABLE FOR SOURCE TYPE 'KTYPE'
C
      DO 90 IRANK=1,4
      IF (CONC  .GT. HRMAX (M,IRANK,KTYPE)) GO TO 95
90    CONTINUE
      IRANK = 5
      GO TO 250
C       PUSH - DOWN THE STACK, STARTING AT IRANK
95    DO 111 LJ=IRANK,4
      LK=5 + IRANK - LJ
      LKM1 = LK - 1
      MAXHR (M,LK,1,KTYPE)=MAXHR(M,LKM1,1,KTYPE)
      MAXHR (M,LK,2,KTYPE)=MAXHR(M,LKM1,2,KTYPE)
      HRMAX (M,LK,KTYPE)= HRMAX (M,LKM1,KTYPE)
111   CONTINUE
250   HRMAX (M,IRANK,KTYPE)=CONC
      MAXHR (M,IRANK,1,KTYPE)=100*KREC + KSRC
      MAXHR (M,IRANK,2,KTYPE)=1000*KHR + IDAYNO
C
C
C===  FIRST,   WHAT CATCHMENTS ARE IN THIS RECEPTOR BOX?
C
500   CALL INDXER (KATCH,KN,JJ(1),JJ(2),JJ(3))
C
C=== ***********************************************************
C===   PUT DEBUG PRINTS HERE.....
       IF (M .EQ. 1) WRITE (LUNDBG,8001) KTYPE,KATCH,KREC,KSRC,KHR
C=== ***********************************************************
      DO 926 LX=1,KN
      JX=JJ(LX)
C
      CONCEN = UWDEPL (JX,M,CONC,TTIME,VTHICK)
      PPTCON (M,JX,KHR) = PPTCON (M,JX,KHR) + CONC
C
      DPLCON (M,JX,KHR) = DPLCON (M,JX,KHR) + CONCEN
C
      DEPOS = FLUX (M, CONCEN) * 3600.0
C
```

```
C   MULTIPLY THE SOURCE-RECEPTOR- 'HOURLY' 'FLUX' BY 3600 (THE NUMBER   UPD00580
C     OF SECONDS IN ONE HOUR)                                           UPD00590
C                                                                       UPD00600
      DAYDEP(M,KREC) = DAYDEP(M,KREC) + DEPOS                           UPD00610
C                                                                       UPD00620
C     'ACCUMULATE ' THE CONCENTRATIONS IN PPTCON...WHETHER THESE GET USEUPD00630
C       OR NOT WILL BE DETERMINED BY THE DAYEND AND OUTPUT ROUTINES, BASUPD00640
C         ON WHETHER THIS CATCHMENT RECEIVED ANY RAINFALL OR NOT FOR THE UPD00650
C     DAY: IF IT DID, HOURLY NON-ZERO CONC'S WILL BE MULTIPLIED BY THE  UPD00660
C     POLLUTANT-SPECIFIC SCAVENGING RATIO ONLY FOR HOURS W/ NON-ZERO PPTUPD00670
C                                                                       UPD00680
C=== ************************************************************       UPD00690
C===  PUT DEBUG PRINTS HERE.....                                        UPD00700
      WRITE (LUNDBG,8002) M,CONC,CONCEN,DEPOS,DAYDEP(M,KREC)            UPD00710
      WRITE (LUNDBG,8003) JX,M,KHR,PPTCON(M,JX,KHR),DPLCON(M,JX,KHR)    UPD00720
C=== ************************************************************       UPD00730
C                                                                       UPD00740
 926  CONTINUE                                                          UPD00750
C                                                                       UPD00760
C                                                                       UPD00770
1000  CONTINUE                                                          UPD00780
C                                                                       UPD00790
100   IF (.NOT. IDBUG) RETURN                                           UPD00800
C=== ************************************************************       UPD00810
C===  PUT DEBUG PRINTS HERE.....                                        UPD00820
8001  FORMAT ('0* UPDATE INDICES: KTYP,KATCH,KR,KS,KH:',5I7)            UPD00830
8002  FORMAT (' * UPDATE  VALUES: M,CONC,CONCEN,DEPOS,DAYDEP(M,KREC):', UPD00840
     X    I3,1P4G12.3)                                                  UPD00850
8003  FORMAT (' * CUM.VAL. OF PPTCON & DPLCON IN KATCH:',I3,            UPD00860
     X        ' FOR POLL.:',I3,' HOUR:',I3,' IS:',1P2G12.3)             UPD00870
C=== ************************************************************       UPD00880
C                                                                       UPD00890
      RETURN                                                            UPD00900
      END                                                               UPD00910
      SUBROUTINE ARCOMP (KHR,KSRC,IX,JY,IA,QN,ERATE,QC,SPLIT,TIMMAX)
      COMMON /ORIGIN/ X0,Y0,UNIT,LX,LY,NPOLLS
      COMMON /RECEPS/ INDEX(225),INDPPT(20),KATCH(20),
     X                NEXAMS(10),KNTEX
      COMMON /VALUES/ ADJWD(2),TAU(2),
     X                UDEPOS(2),USETTL(2),SCAVNG(2),BKGRND(2),RUFLEN
      COMMON /ARANGL/ EPS(24),KEPS(24),ISIGNX(24),ISIGNY(24),
     X                ITABX(24),ITABY(24),INCRX(24),INCRY(24)
      COMMON /ACONST/ EPSLIM(10),AAR(6),BAR(6),ITABLE(5,10)
      COMMON /METDAT/ IDAYS(366),THETA(24),U(24),KST(24),HL(24),
     X                TEMP(24),IRAIN(25,5)
      COMMON /CHITAB/ HRMAXP(2,5),HRMAXA(2,5),HRMAXC(2,5),
     X                MAXHRP(2,5,2),MAXHRA(2,5,2),MAXHRC(2,5,2),
     X                DAYMAX(2,10)
      COMMON /MCONST/ KNTDAY(12),NDAY,IYEAR,IMONTH,IDAYNO,LL(5)
C
C=== ******************************************************************
      COMMON / TEST / IDBUG,LUNDBG
      LOGICAL IDBUG
C=== ******************************************************************
C
      DIMENSION  QN(5), ERATE(2)
C
C
C  LOOP FOR THE FIVE RECEPTORS AFFECTED BY EACH RECEPTOR GRID SQUARE
C    CONTAINING EMISSIONS.  THE RECEPTORS IN THE FOUR DOWNWIND SQUARES
C    AND IN THE SOURCE SQUARE ITSELF ARE AFFECTED.
C
      IEPS=KEPS(KHR)
      DO 2426 L=1,5
      NX= IX + ISIGNX(KHR)*(ITABLE(L,IEPS)*ITABX(KHR) +
     X (L-1)*INCRX(KHR)) - 4
      IF(NX.LT.1.OR.NX.GT.LX) GO TO 2426
      NY= JY + ISIGNY(KHR)*(ITABLE(L,IEPS)*ITABY(KHR) +
     X (L-1)*INCRY(KHR)) - 4
      IF(NY.LT.1.OR.NY.GT.LY) GO TO 2426
      NCELL=(NX-1)*LY+NY
      KATID=INDEX(NCELL)
C
      DO 2425 M=1,NPOLLS
      ARCON = 0.0
      DECAY = 0.0
      IF (ERATE(M) .EQ. 0.0) GO TO 2425
      HLIVES = TIMMAX/TAU(M)
      IF (HLIVES - 50.0) 2424, 2425, 2425
2424  DECAY = EXP(- 0.6931472 * HLIVES)
C
C  CALCULATE CONCENTRATION OF POLLUTANT M AT RECEPTOR IN COLUMN NX,
C    ROW NY OF THE RECEPTOR GRID DUE TO AREA SOURCE KSRC.
C
      ARCON= QC*QN(L)*SPLIT*ERATE(M)*DECAY
C
      CALL  UPDATE (M,2,NCELL,KSRC,KATID,KHR,ARCON,TIMMAX,HL(KHR))
C
100   IF (.NOT. IDBUG) GO TO 2425
C=== ******************************************************************
C===  PUT DEBUG PRINTS HERE.....
```

```
      IF (M .EQ. 1) WRITE (LUNDBG,999) KHR,L,NX,NY,NCELL,DECAY,ARCON    ARC00580
999   FORMAT (2X,'* ARCOMP: KHR,L,NX,NY,NCELL,D-K,ARCON=',              ARC00590
     X        2X,5I3,1P2G11.3)                                          ARC00600
C=== **************************************************************     ARC00610
C                                                                       ARC00620
2425  CONTINUE                                                          ARC00630
2426  CONTINUE                                                          ARC00640
      RETURN                                                            ARC00650
      END                                                               ARC00660
      FUNCTION SYCOMP(X,ICL)                                            NUD21050
C     ROUTINE TO COMPUTE SY USING TABLE LOOK-UP                         NUD21060
C     PARAMETERS:                                                       NUD21070
C     X--DOWNWIND DISTANCE (METERS)                                     NUD21080
C     ICL--STABILITY CLASS (ASME 4 LEVEL)                               NUD21090
      COMMON /TAB001/ HLOW,HHIGH,HDEL,XLTAB(50,4),                      NUD21100
     X                XLOW,XHIGH,XDEL,SYTAB(50,4),SZTAB(50,4),          NUD21110
     X                ELOW,EMAX,EDEL,ETAB(100),NCLASS,NHT,NXC,NEXP      NUD21120
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,                       NUD21130
     X                T,P,SYCOEF(4),SYEXP(4),SZCOEF(4),SZEXP(4),        NUD21140
     X                R2PI,ROOT2,ROOTPI,                                NUD21150
     X                J1STAB(7),J2STAB(7)                               NUD21160
C                                                                       NUD21170
      IF(X.GE.XLOW.AND.X.LT.XHIGH) GO TO 25                             NUD21180
      SYCOMP=SYCOEF(ICL)*X**SYEXP(ICL)                                  NUD21190
      RETURN                                                            NUD21200
25    RATIO=(X-XLOW)/XDEL                                               NUD21210
      ID=IFIX(RATIO)+1                                                  NUD21220
C     GET DISTANCE CLASS                                                NUD21230
      DELTA=RATIO-IFIX(RATIO)                                           NUD21240
      SYCOMP=SYTAB(ID,ICL)+DELTA*(SYTAB(ID+1,ICL)-SYTAB(ID,ICL))        NUD21250
      RETURN                                                            NUD21260
      END                                                               NUD21270
      FUNCTION SZCOMP(X,ICL)                                            NUD21280
C     ROUTINE TO COMPUTE SZ USING TABLE LOOK-UP                         NUD21290
C     PARAMETERS:                                                       NUD21300
C     X--DOWNWIND DISTANCE (METERS)                                     NUD21310
C     ICL--STABILITY CLASS (ASME 4 LEVEL)                               NUD21320
      COMMON /TAB001/ HLOW,HHIGH,HDEL,XLTAB(50,4),                      NUD21330
     X                XLOW,XHIGH,XDEL,SYTAB(50,4),SZTAB(50,4),          NUD21340
     X                ELOW,EMAX,EDEL,ETAB(100),NCLASS,NHT,NXC,NEXP      NUD21350
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,                       NUD21360
     X                T,P,SYCOEF(4),SYEXP(4),SZCOEF(4),SZEXP(4),        NUD21370
     X                R2PI,ROOT2,ROOTPI,                                NUD21380
     X                J1STAB(7),J2STAB(7)                               NUD21390
C                                                                       NUD21400
      IF(X.GE.XLOW.AND.X.LT.XHIGH) GO TO 25                             NUD21410
      SZCOMP=SZCOEF(ICL)*X**SZEXP(ICL)                                  NUD21420
      RETURN                                                            NUD21430
   25 RATIO=(X-XLOW)/XDEL                                               NUD21440
      ID=IFIX(RATIO)+1                                                  NUD21450
C     GET DISTANCE CLASS                                                NUD21460
      DELTA=RATIO-IFIX(RATIO)                                           NUD21470
      SZCOMP=SZTAB(ID,ICL)+DELTA*(SZTAB(ID+1,ICL)-SZTAB(ID,ICL))        NUD21480
      RETURN                                                            NUD21490
      END                                                               NUD21500
      FUNCTION EXCOMP(X)                                                NUD13730
C     FUNCTION TO EVALUATE EXP(-.5*X**2)                                NUD13740
      COMMON /TAB001/ HLOW,HHIGH,HDEL,XLTAB(50,4),                      NUD13750
     X                XLOW,XHIGH,XDEL,SYTAB(50,4),SZTAB(50,4),          NUD13760
     X                ELOW,EMAX,EDEL,ETAB(100),NCLASS,NHT,NXC,NEXP      NUD13770
C                                                                       NUD13780
      IF(ABS(X).LE.EMAX) GO TO 10                                       NUD13790
      EXCOMP=0.0                                                        NUD13800
      RETURN                                                            NUD13810
10    RATIO=(ABS(X)-ELOW)/EDEL                                          NUD13820
      ID=IFIX(RATIO)+1                                                  NUD13830
      DELTA=RATIO-IFIX(RATIO)                                           NUD13840
      EXCOMP=ETAB(ID)+DELTA*(ETAB(ID+1)-ETAB(ID))                       NUD13850
      RETURN                                                            NUD13860
      END                                                               NUD13870
      FUNCTION XLCOMP(HL,ICL)                                           NUD24320
C     ROUTINE TO COMPUTE DISTANCE XL S.T. SZ(XL) = .47 * HL             NUD24330
C     PARAMETERS:                                                       NUD24340
C     HL--MIXING HEIGHT (M)                                             NUD24350
C     ICL--STABILITY CLASS (ASME 4 LEVEL)                               NUD24360
      COMMON /TAB001/ HLOW,HHIGH,HDEL,XLTAB(50,4),                      NUD24370
     X                XLOW,XHIGH,XDEL,SYTAB(50,4),SZTAB(50,4),          NUD24380
     X                ELOW,EMAX,EDEL,ETAB(100),NCLASS,NHT,NXC,NEXP      NUD24390
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,                       NUD24400
     X                T,P,SYCOEF(4),SYEXP(4),SZCOEF(4),SZEXP(4),        NUD24410
     X                R2PI,ROOT2,ROOTPI,                                NUD24420
     X                J1STAB(7),J2STAB(7)                               NUD24430
C                                                                       NUD24440
C=== ****************************************************************** NUD24450
      COMMON / TEST / IDBUG,LUNDBG                                      NUD24460
      LOGICAL IDBUG                                                     NUD24470
C=== ****************************************************************** NUD24480
C                                                                       NUD24490
C                                                                       NUD24500
      IF(HL.GE.HLOW.AND.HL.LT.HHIGH) GO TO 25                           NUD24510
      XLCOMP=(0.47*HL/SZCOEF(ICL))**(1.0/SZEXP(ICL))                    NUD24520
      RETURN                                                            NUD24530
25    CONTINUE                                                          NUD24540
      RATIO=(HL-HLOW)/HDEL                                              NUD24550
      IH=IFIX(RATIO)+1                                                  NUD24560
C     GET HEIGHT CLASS                                                  NUD24570
      DELTA=RATIO-IFIX(RATIO)                                           NUD24580
      XLCOMP=XLTAB(IH,ICL)+DELTA*(XLTAB(IH+1,ICL)-XLTAB(IH,ICL))        NUD24590
C     XL IS IN METERS                                                   NUD24600
C     COMPUTE DISTANCE S.T. SZ(XL) = .47*HL                             NUD24610
C                                                                       NUD24620
100   IF (.NOT. IDBUG) RETURN                                           NUD24630
C=== ****************************************************************** NUD24640
C===  PUT DEBUG PRINTS HERE.....                                        NUD24650
C=== ****************************************************************** NUD24660
C                                                                       NUD24670
      RETURN                                                            NUD24680
      END                                                               NUD24690
      FUNCTION CERMAK(Y,SY,H,SZ,KZ,V,W)
C     ROUTINE TO EVALUATE ERMAK EXPRESSION FOR DOWNWIND CONCENTRATION
C     LIMITED TO GROUND LEVEL RECEPTORS (Z = 0)
C     PARAMETERS:
C     Y--CROSSWIND DISTANCE (M)
C     SY--CROSSWIND DISPERSION (M)
C     H--EFFECTIVE SOURCE HEIGHT (M)
C     SZ--VERTICAL DISPERSION (M)
C     KZ--VERTICAL DIFFUSION COEFFICIENT (M**2/S)
C     V--DEPOSITION VELOCITY (M/S)
C     W--GRAVITATIONAL SETTLING VELOCITY (M/S)
      REAL KZ
C
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,
     X                T,P,YCOEFF(4),YPOWER(4),ZCOEFF(4),ZPOWER(4),
     X                R2PI,ROOT2,ROOTPI,
     X                J1STAB(7),J2STAB(7)
C
C
C=== ***********************************************************
      COMMON / TEST / IDBUG,LUNDBG
      LOGICAL IDBUG
C=== ***********************************************************
C
1     V1=V-0.5*W
      HCOEF=H/SZ
      TERM1=EXCOMP(Y/SY)/(ROOTPI*SY)
C     COMPUTE CROSSWIND TERM (GAUSSIAN)
5     TEMPA=(W/(2.0*KZ))*(H-W*SZ**2/(4.0*KZ))-0.5*HCOEF**2
      IF(TEMPA .LT.-75.0) TEMPA=-75.0
      IF(TEMPA .GT. 75.0) TEMPA= 75.0
      TERM2=EXP(TEMPA)/(ROOTPI*SZ)
      ERR1=ERFC((V1*SZ/KZ+HCOEF)/ROOT2)
      TERM3=0.0
      IF(ERR1.LE.0.0) GO TO 10
      TEMPB=(V/KZ)*(H+SZ**2*(V-W)/(2.0*KZ))
      TLOG1=ALOG(ERR1)
      TEMPQ =TLOG1+TEMPB
      IF(TEMPQ .LT.-75.0) TEMPQ=-75.0
      IF(TEMPQ .GT. 75.0) TEMPQ= 75.0
      TERM3=(V1/(ROOT2*KZ))*EXP(TEMPQ)
10    CONTINUE
      CERMAK=TERM1*(TERM2-TERM3)
      IF (CERMAK .LT. 1.0E-35) CERMAK = 0.0
100   IF (.NOT. IDBUG) RETURN
C
C=== ***********************************************************
C=== PUT DEBUG PRINTS HERE.....
      FCNVAL = CERMAK
      WRITE (LUNDBG,999) FCNVAL,Y,SY,H,SZ,KZ,V,W
999   FORMAT (2X,'* CERMAK: * FCNVAL,Y,SY,H,SZ,KZ,V,W=',
     X      1P8G10.2)
C=== ***********************************************************
C
      RETURN
      END
      FUNCTION DEPRAT(H,SZ,KZ,V,W)                                      DEP00010
C     ROUTINE TO COMPUTE NET DEPOSITION RATE FRACTION                   DEP00020
C     PARAMETERS:                                                       DEP00030
C     H--SOURCE HEIGHT (M)                                              DEP00040
C     SZ--VERTICAL DISPERSION (M)                                       DEP00050
C     KZ--VERTICAL DIFFUSION COEFFICIENT (M**2/S)                       DEP00060
C     V--DEPOSITION VELOCITY (M/S)                                      DEP00070
C     W--GRAVITATIONAL SETTLING VELOCITY (M/S)                          DEP00080
      REAL KZ                                                           DEP00090
C                                                                       DEP00100
      COMMON /PCONST/ WSPDS(10),WIDTH(7),G,PION4,                       DEP00110
     X                T,P,YCOEFF(4),YPOWER(4),ZCOEFF(4),ZPOWER(4),      DEP00120
     X                R2PI,ROOT2,ROOTPI,                                DEP00130
     X                J1STAB(7),J2STAB(7)                               DEP00140
C                                                                       DEP00150
C=== ***************************************************************    DEP00160
      COMMON / TEST / IDBUG,LUNDBG                                      DEP00170
      LOGICAL IDBUG                                                     DEP00180
C=== ***************************************************************    DEP00190
C                                                                       DEP00200
      IF(V.EQ.W) GO TO 25                                               DEP00210
C     CHECK FOR INDETERMINATE CASE                                      DEP00220
      WCOEF=W*SZ/KZ                                                     DEP00230
      V1=V-0.5*W                                                        DEP00240
      V2=V-W                                                            DEP00250
      V1COEF=V1*SZ/KZ                                                   DEP00260
      V2COEF=V2*SZ/KZ                                                   DEP00270
      HCOEF=H/SZ                                                        DEP00280
      TEMPA=(WCOEF/2.0-HCOEF)/ROOT2                                     DEP00290
      TERM1=0.5*ERFC(TEMPA)                                             DEP00300
      ERR1=ERFC((HCOEF+0.5*WCOEF)/ROOT2)                                DEP00310
      TERM2=0.0                                                         DEP00320
      IF(ERR1.LE.0.0) GO TO 10                                          DEP00330
      TLOG1=W*H/KZ+ALOG(ERR1)                                           DEP00340
      TERM2=(V/(2.0*V2))*EXP(TLOG1)                                     DEP00350
10    CONTINUE                                                          DEP00360
      ERR2=ERFC((HCOEF+V1COEF)/ROOT2)                                   DEP00370
      TERM3=0.0                                                         DEP00380
      IF(ERR2.LE.0.0) GO TO 15                                          DEP00390
      TLOG2=(V/KZ)*(H+0.5*V2COEF*SZ)+ALOG(ERR2)                         DEP00400
      TERM3=(V1/V2)*EXP(TLOG2)                                          DEP00410
15    CONTINUE                                                          DEP00420
      DEPRAT=1.0-TERM1+TERM2-TERM3                                      DEP00430
      GO TO 100                                                         DEP00440
25    CONTINUE                                                          DEP00450
C*****     SPECIAL CASE V = W      *****                                DEP00460
      VCOEF=V*SZ/KZ                                                     DEP00470
      HCOEF=H/SZ                                                        DEP00480
      TEMPA=(VCOEF/2.0-HCOEF)/ROOT2                                     DEP00490
      TERM1=0.5*ERFC(TEMPA)                                             DEP00500
      TERM2=(VCOEF/R2PI)*EXP(-TEMPA**2)                                 DEP00510
      TEMPB=1.0+0.5*VCOEF**2+V*H/KZ                                     DEP00520
      ERR1=ERFC((HCOEF+0.5*VCOEF)/ROOT2)                                DEP00530
      TERM3=0.0                                                         DEP00540
      IF(ERR1.LE.0.0) GO TO 30                                          DEP00550
      TLOG1=V*H/KZ+ALOG(ERR1)                                           DEP00560
      TERM3=0.5*TEMPB*EXP(TLOG1)                                        DEP00570
```

```
30    CONTINUE                                                          DEP00580
      DEPRAT=1.0-TERM1+TERM2-TERM3                                      DEP00590
C                                                                       DEP00600
100   IF (.NOT. IDBUG) RETURN                                           DEP00610
C=== *************************************************************      DEP00620
C===  PUT DEBUG PRINTS HERE.....                                        DEP00630
      FCNVAL=DEPRAT                                                     DEP00640
      WRITE (LUNDBG,800) FCNVAL,TERM1,TERM2,TERM3,V,W                   DEP00650
800   FORMAT (3X,'***** DEPRAT: * VALUE,TERMS,V,W:',2X,1P6G12.2)        DEP00660
C=== *************************************************************      DEP00670
C                                                                       DEP00680
      RETURN                                                            DEP00690
      END                                                               DEP00700
      SUBROUTINE DAYUPD (M,KATCH,CONC)
      COMMON /CHITAB/ HRMAX(2,5,3),
     X                MAXHR(2,5,2,3),
     X                DAYMAX(2,10),MAXDAY(2,10)
      COMMON /MCONST/ KNTDAY(12),NDAY,IYEAR,IMONTH,IDAYNO,L(5)
C
C     THIS SUBROUTINE  HANDLES CATCHMENT-AVERAGE CONC. RANKING
C
C
C       COMPARE CONC ARRAY TO THE CURRENT 10 HIGHEST DAILY CONCENTRATION
C       VALUES FOR ALL SOURCE TYPES AT ALL CATCHMENTS FOR THIS DAY
C
      IF (CONC .LT. DAYMAX (M,10)) RETURN
C
C      THIS VALUE OF CONC  BELONGS IN THE TABLE FOR THIS DAY
C
      DO 190 IRANK=1,9
      IF (CONC  .GT. DAYMAX (M,IRANK)) GO TO 195
  190  CONTINUE
      IRANK = 10
      GO TO 1250
C       PUSH - DOWN THE STACK, STARTING AT IRANK
  195 DO 1100 LJ=IRANK,9
      LK=10 + IRANK - LJ
      LKM1 = LK - 1
      MAXDAY (M,LK)=MAXDAY(M,LKM1)
      DAYMAX (M,LK)= DAYMAX (M,LKM1)
 1100  CONTINUE
 1250 DAYMAX (M,IRANK)=CONC
      MAXDAY (M,IRANK)=1000*KATCH + IDAYNO
C
      RETURN
      END
      SUBROUTINE OUTPUT(AVFLUX,IRON)                                    NUD16960
      COMMON /ORIGIN/ X0,Y0,UNIT,NEAST,NNORTH,NPOLLS                    NUD16970
      COMMON /RECEPS/ INDEX(225),INDPPT(20),KATCH(20),                  NUD16980
     X                NEXAMS(10),KNTEX                                  NUD16990
      COMMON /DISKOP/ DAYDEP(2,225),PPTCON(2,20,24)                     NUD16990
      COMMON /VALUES/ ADJWD(2),TAU(2),                                  NUD17000
     X                UDEPOS(2),USETTL(2),SCAVNG(2),BKGRND(2),RUFLEN    NUD17010
      COMMON /IODEVS/ LUNIN,LUNPRT,LUNPUN,LUNMET,LUNNPS,NPPT,LUNPPT(5)  NUD17020
      COMMON /MCONST/ KNTDAY(12),NDAY,IYEAR,IMONTH,IDAYNO,LX(5)         NUD17030
      COMMON /METDAT/ IDAYS(366),THETA(24),U(24),KST(24),HL(24),        NUD17040
     X                TEMP(24),IRAIN(25,5),TSNOW                        NUD17050
      DIMENSION AVFLUX(2,20)                                            NUD17060
C                                                                       NUD17070
 1000 IF (IRON .EQ. 0) GO TO 7000                                       NUD17080
C                                                                       NUD17090
      DO 2000 L = 1,20                                                  NUD17100
      JRAIN = INDPPT(L)-60                                              NUD17110
      IF (JRAIN .GT. 5 .OR. JRAIN .LT. 1) GO TO 2000                    NUD17120
      AREA = FLOAT( KATCH(L) )                                          NUD17130
      IF (AREA .EQ. 0.0) GO TO 2000                                     NUD17140
C                                                                       NUD17150
C     NOW WE WILL CONVERT THE CONTENTS OF PPTCON ARRAY FROM CONCEN-     NUD17160
C     TRATIONS IN THE AIR BY HOUR BY POLLUTANT                          NUD17170
C     INTO 'WET DEPOSITION FLUX'  VALUES BY POLLUTANT ONLY              NUD17180
C     FOR THOSE HOURS ON WHICH THERE WAS RAIN ON THIS DAY..OTHER HOURS  NUD17190
C     WILL TURN OUT TO BE ZERO.                                         NUD17200
C     NOTE ALSO THAT IF THERE WAS PRECIPITATION AND THE AMBIENT TEMP.   NUD17210
C     IS BELOW 'TSNOW', THE PRECIPITATION IS ASSUMED TO BE FROZEN, AND  NUD17220
C     ITS ASSOCIATED FLUX WILL BE ADDED TO THE DAILY DRY DEP. FLUX      NUD17230
C     FOR THE DAY                                                       NUD17240
C                                                                       NUD17250
      DO 1500 KHR = 1,24                                                NUD17260
      FLAG1=1.0                                                         NUD17270
      IF (IRAIN(KHR,JRAIN) .EQ. 0) FLAG1=0.0                            NUD17280
C                                                                       NUD17290
      DO 1200 M=1,NPOLLS                                                NUD17300
      PPTCON(M,L,KHR)=FLAG1 *  PPTCON(M,L,KHR) * SCAVNG(M) *            NUD17310
     +          IRAIN(KHR,JRAIN) * 2.54 E -4  / AREA                    NUD17320
C                                                                       NUD17330
C                                                                       NUD17340
       IF (FLAG1 .EQ. 0.0 .OR.(FLAG1.EQ.1.0 .AND. TEMP(KHR) .GE. TSNOW)) NUD17350
     X          GO TO 1200                                              NUD17360
      AVFLUX(M,L) = AVFLUX(M,L) + PPTCON(M,L,KHR) * 3600.0              NUD17370
      PPTCON(M,L,KHR) = 0.0                                             NUD17380
C                                                                       NUD17390
 1200 CONTINUE                                                          NUD17400
C                                                                       NUD17410
C   THE CONSTANT 2.54 E-4 CONVERTS THE RAINFALL AMOUNTS FROM HUNDREDTHS NUD17420
C  OF AN INCH TO METERS, SO THE UNITS OF THE RESULT ARE MICROGRAMS/M**2,NUD17430
C   AN HOURLY FLUX  BY NPS AND EXAMS STANDARDS                          NUD17440
C                                                                       NUD17450
1500  CONTINUE                                                          NUD17460
C                                                                       NUD17470
 2000 CONTINUE                                                          NUD17480
C                                                                       NUD17490
 7000 CONTINUE                                                          NUD17500
      KRTYPE = 2                                                        NUD17510
                                                                        NUD17520
      WRITE (LUNNPS) KRTYPE                                             NUD17530
      WRITE (LUNNPS) IYEAR,IMONTH,IDAYNO,IRON,                          NUD17540
     X          (M,(AVFLUX(M,N),N=1,20),M=1,NPOLLS)                     NUD17550
C                                                                       NUD17560
C                                                                       NUD17570
      IF (IRON .EQ. 0) RETURN                                           NUD17580
      KRTYPE = 3                                                        NUD17590
C                                                                       NUD17600
      DO 8000 L = 1,20                                                  NUD17610
      IF (KATCH (L) .EQ. 0) GO TO 8000                                  NUD17620
      WRITE (LUNNPS) KRTYPE                                             NUD17630
      WRITE (LUNNPS) IYEAR,IMONTH,IDAYNO,L,                             NUD17640
     X        (M,(PPTCON(M,L,KHR),KHR=1,24),M=1,NPOLLS)                 NUD17650
8000  CONTINUE                                                          NUD17660
C                                                                       NUD17670
C                                                                       NUD17680
      RETURN                                                            NUD17690
      END                                                               NUD17700
      FUNCTION UWDEPL (JX,M,CONC,TTIME,VTHICK)                          UWD00010
      COMMON /RECEPS/ INDEX(225),INDPPT(20),KATCH(20),                  UWD00020
     X                NEXAMS(10),KNTEX                                  UWD00030
      COMMON /METDAT/ IDAYS(366),ANGL(24),U(24),KST(24),HL(24),TEMP(24),UWD00040
     X                IRAIN(25,5),TSNOW                                 UWD00050
      COMMON /VALUES/ ADJWD(2),TAU(2),                                  UWD00060
     X                UDEPOS(2),USETTL(2),SCAVNG(2),BKGRND(2),RUFLEN    UWD00070
C                                                                       UWD00080
C=== ***********************************************************        UWD00090
      COMMON / TEST / IDBUG,LUNDBG                                      UWD00100
      LOGICAL IDBUG                                                     UWD00110
C=== ***********************************************************        UWD00120
C                                                                       UWD00130
C===  BY DEFAULT (I.E. NO RAIN), THEN NO UPWIND WET DEPLETION AND..     UWD00140
C                                                                       UWD00150
      UWDEPL = CONC                                                     UWD00160
      JRAIN = 0                                                         UWD00170
C                                                                       UWD00180
C===  WAS THERE ANY RAIN FOR THIS HOUR IN THE CATCHMENT INDEXED BY 'JX'?UWD00190
C                                                                       UWD00200
      IUNPPT = INDPPT (JX) - 60                                         UWD00210
      IF (IUNPPT .GT. 5 .OR. IUNPPT .LT. 1) GO TO 1000                  UWD00220
C                                                                       UWD00230
C===  IF IUNPPT NOT 1 THRU 5, NO PRECIP. FILE ASSOCIATED WITH THIS CATCHUWD00240
C                                                                       UWD00250
C===  ANY RAIN THIS HOUR?                                               UWD00260
C                                                                       UWD00270
      JRAIN = IRAIN(KHR,IUNPPT)                                         UWD00280
      IF (JRAIN .EQ. 0) GO TO 1000                                      UWD00290
C                                                                       UWD00300
C===  UPWIND DEPL. BY WET SCAVNGNG IS APPROX'D BY EXP (- K * T), WHERE: UWD00310
C===     T = 'TRAVEL TIME' BETWEEN SOURCE AND RECEPTOR (IN HRS.)        UWD00320
C===     K = SCAVENGING RATE:                                           UWD00330
C===           K = (SCAVNGNG RATIO) * (PRECIP. RATE) / (EFFECTIVE PLUME UWD00340
C===                                                        THICKNESS  )UWD00350
C===                 SCAVNGNG RATIO IS 'DIMENSIONLESS'                  UWD00360
C===                 PRECIP. RATE IS 'JRAIN' (IN 0.01 INCHES/HOUR)      UWD00370
C===                 EFFECTIVE PLUME THICKNESS IS VTHICK (IN METERS)    UWD00380
C===                                                                    UWD00390
C                                                                       UWD00400
      DPLETE = TTIME * (SCAVNG(M) * JRAIN * 2.54E-4) / VTHICK           UWD00410
C                                                                       UWD00420
      IF (DPLETE .GT. 30.) DPLETE = 30.                                 UWD00430
C                                                                       UWD00440
C=== ***********************************************************        UWD00450
C===  PUT DEBUG PRINTS HERE.....                                        UWD00460
      IF (M .EQ. 1) WRITE (LUNDBG,8001) JRAIN,TTIME,                    UWD00470
     X                                 VTHICK                           UWD00480
C=== ***********************************************************        UWD00490
      UWDEPL = CONC * EXP (-DPLETE)                                     UWD00500
C                                                                       UWD00510
C=== ***********************************************************        UWD00520
1000  CONTINUE                                                          UWD00530
C                                                                       UWD00540
100   IF (.NOT. IDBUG) RETURN                                           UWD00550
C=== ***********************************************************        UWD00560
C===  PUT DEBUG PRINTS HERE.....                                        UWD00570
```

```
      FCNVAL = UWDEPL                                                   UWD00580
      WRITE (LUNDBG,8002) M,FCNVAL,CONC,DPLETE                          UWD00590
C=== **************************************************************    UWD00600
8001  FORMAT (' * UWDEPL INDICES: JRAIN,TTIME,VTHICK:',I4,1P2G12.2)     UWD00610
8002  FORMAT (' * UWDEPL  VALUES: M,FCNVAL,CONC,DPLETE:',I3,            UWD00620
     X        1P3G12.3)                                                 UWD00630
C=== **************************************************************    UWD00640
C                                                                       UWD00650
      RETURN                                                            UWD00660
      END                                                               UWD00670
```

DIDOT POSTPROCESSOR

```
C                                                                       MAI00010
C  THE MAIN ROUTINE DETERMINES THE RECORD TYPE, AND THEN CALLS THE      MAI00020
C  SUBROUTINE APPROPRIATE TO DEAL WITH EACH RECORD TYPE.                MAI00030
C                                                                       MAI00040
C     IMPLICIT INTEGER*2 (I-N)                                          MAI00050
      INTEGER LUNPRT, LUNPIN, LUNNPS, LUNEXM, ITEMP                     MAI00060
      COMMON /FLAGS/ ITYPE, IFLAG, IYEAR, IMONTH, NDAYS, IYRCT          MAI00070
      COMMON /IO/ LUNPRT, LUNPIN, LUNNPS, LUNEXM                        MAI00080
      DIMENSION LUNNPS(20)                                              MAI00090
      CALL INIT                                                         MAI00100
C                                                                       MAI00110
C                                                                       MAI00120
    1 READ( LUNPIN, END = 100 ) ITYPE                                   MAI00130
C                                                                       MAI00140
      IF( ITYPE .NE. 1 ) GOTO 20                                        MAI00150
         CALL REC1                                                      MAI00160
         GOTO 1                                                         MAI00170
C                                                                       MAI00180
   20 IF( ITYPE .NE. 2 ) GOTO 30                                        MAI00190
         CALL REC2                                                      MAI00200
         GOTO 1                                                         MAI00210
C                                                                       MAI00220
   30 IF( ITYPE .NE. 3 ) GOTO 999                                       MAI00230
         CALL REC3                                                      MAI00240
         GOTO 1                                                         MAI00250
C                                                                       MAI00260
  100 CALL PRTPOT                                                       MAI00270
      CALL PRTTSP                                                       MAI00280
      STOP                                                              MAI00290
C                                                                       MAI00300
  999 WRITE( LUNPRT, 1000 ) ITYPE                                       MAI00310
 1000 FORMAT( ' UNKNOWN RECORD TYPE, ', I4, '. PROGRAM HALTED.' )       MAI00320
      STOP                                                              MAI00330
      END                                                               MAI00340
      BLOCK DATA                                                        BL000010
C     IMPLICIT INTEGER*2 (I-N)                                          BL000020
      INTEGER LUNPRT, LUNPIN, LUNNPS, LUNEXM, LUNUIN                     BL000030
      COMMON /IO/ LUNPRT, LUNPIN, LUNNPS, LUNEXM                        BL000040
      COMMON /FLAGS/ ITYPE, IFLAG, IYEAR, IMONTH, NDAYS, IYRCT          BL000050
      COMMON /POTEN/ PFACIN, IPFLAG                                     BL000060
      COMMON /UIO/ LUNUIN                                               BL000070
      COMMON /MATS/ AFLUX, PFAC, DOTRO1, RFLUX, DOTRO2, FLXSUM, TSPMAT  BL000080
      COMMON /POLKAT/ NPOLLS, KATDEF                                    BL000090
      COMMON /WTMATS/ WETFLX, EXMWET                                    BL000100
      DIMENSION LUNNPS(20)                                              BL000110
      DIMENSION PFACIN(2,12,5)                                          BL000120
      DIMENSION AFLUX(12,31), PFAC(2), DOTRO1(10), RFLUX(24,3),         BL000130
     *          DOTRO2(10), FLXSUM(3,20), TSPMAT(20,12,31)              BL000140
      DIMENSION KATDEF(20)                                              BL000150
      DIMENSION WETFLX(3,24), EXMWET(20,3)                              BL000160
      DATA LUNNPS /21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,     BL000170
     *          37,38,39,40/, LUNEXM /41/, LUNPRT /6/, LUNUIN /15/,     BL000180
     *          LUNPIN /16/                                             BL000190
      END                                                               BL000200
      SUBROUTINE DAYMON( IDAYNO, IYR, IMONTH, IDAY )                    DAY00010
C                                                                       DAY00020
C  TAKING THE DAY NUMBER (IDAYNO: 1 THROUGH 365, OR 366 FOR LEAP YEARS) DAY00030
```

```
C  AND YEAR NUMBER (IYR: TWO OR MORE DIGITS) AS INPUTS, DAYMON RETURNS  DAY00040
C  THE MONTH (IMONTH: 1-12) AND DAY OF THE MONTH (IDAY: 1-28, 29, 30, ORDAY00050
C  31).                                                                 DAY00060
C                                                                       DAY00070
C     IMPLICIT INTEGER*2 (I-N)                                          DAY00080
      INTEGER LUNPRT, LUNPIN, LUNNPS, LUNEXM, ITEMP                     DAY00090
      COMMON /IO/ LUNPRT, LUNPIN, LUNNPS, LUNEXM                        DAY00100
      DIMENSION LUNNPS(10)                                              DAY00110
      IDAY = IDAYNO                                                     DAY00120
C                                                                       DAY00130
C  SET LEAP YEAR FLAG.                                                  DAY00140
C                                                                       DAY00150
      LEAP = 0                                                          DAY00160
      ITEMP = IYR                                                       DAY00170
      IF( MOD(ITEMP,4) .EQ. 0 .AND. MOD(ITEMP,100) .NE. 0 ) LEAP = 1    DAY00180
C                                                                       DAY00190
      IF( IDAY .GT. 31 ) GOTO 2                                         DAY00200
         IMONTH = 1                                                     DAY00210
         RETURN                                                         DAY00220
C                                                                       DAY00230
    2 IDAY = IDAY - 31                                                  DAY00240
      IF( IDAY .GT. (28 + LEAP) ) GOTO 3                                DAY00250
         IMONTH = 2                                                     DAY00260
         RETURN                                                         DAY00270
C                                                                       DAY00280
    3 IDAY = IDAY - 28 - LEAP                                           DAY00290
      IF( IDAY .GT. 31 ) GOTO 4                                         DAY00300
         IMONTH = 3                                                     DAY00310
         RETURN                                                         DAY00320
C                                                                       DAY00330
    4 IDAY = IDAY - 31                                                  DAY00340
      IF( IDAY .GT. 30 ) GOTO 5                                         DAY00350
         IMONTH = 4                                                     DAY00360
         RETURN                                                         DAY00370
C                                                                       DAY00380
    5 IDAY = IDAY - 30                                                  DAY00390
      IF( IDAY .GT. 31 ) GOTO 6                                         DAY00400
         IMONTH = 5                                                     DAY00410
         RETURN                                                         DAY00420
C                                                                       DAY00430
    6 IDAY = IDAY - 31                                                  DAY00440
      IF( IDAY .GT. 30 ) GOTO 7                                         DAY00450
         IMONTH = 6                                                     DAY00460
         RETURN                                                         DAY00470
C                                                                       DAY00480
    7 IDAY = IDAY - 30                                                  DAY00490
      IF( IDAY .GT. 31 ) GOTO 8                                         DAY00500
         IMONTH = 7                                                     DAY00510
         RETURN                                                         DAY00520
C                                                                       DAY00530
    8 IDAY = IDAY - 31                                                  DAY00540
      IF( IDAY .GT. 31 ) GOTO 9                                         DAY00550
         IMONTH = 8                                                     DAY00560
         RETURN                                                         DAY00570
C                                                                       DAY00580
    9 IDAY = IDAY - 31                                                  DAY00590
      IF( IDAY .GT. 30 ) GOTO 10                                        DAY00600
```

```
            IMONTH = 9                                                  DAY00610
            RETURN                                                      DAY00620
C                                                                       DAY00630
   10 IDAY = IDAY - 30                                                  DAY00640
         IF( IDAY .GT. 31 ) GOTO 11                                     DAY00650
            IMONTH = 10                                                 DAY00660
            RETURN                                                      DAY00670
C                                                                       DAY00680
   11 IDAY = IDAY - 31                                                  DAY00690
         IF( IDAY .GT. 30 ) GOTO 12                                     DAY00700
            IMONTH = 11                                                 DAY00710
            RETURN                                                      DAY00720
C                                                                       DAY00730
   12 IDAY = IDAY - 30                                                  DAY00740
         IF( IDAY .GT. 31 ) GOTO 13                                     DAY00750
            IMONTH = 12                                                 DAY00760
            RETURN                                                      DAY00770
C                                                                       DAY00780
C                                                                       DAY00790
   13 WRITE( LUNPRT, 100 ) IDAYNO                                       DAY00800
  100 FORMAT( ' DAY NUMBER, ', I4, ', GREATER THAN 365 (OR 366).',     DAY00810
     *        '  PROGRAM HALTED.' )                                     DAY00820
      STOP                                                              DAY00830
      END                                                               DAY00840
      SUBROUTINE GETPOT                                                 GET00010
C                                                                       GET00020
C  GETPOT READS IN THE MONTHLY BACKGROUND POTENCY FACTORS FOR           GET00030
C  2 POLLUTANTS AND 3 YEARS.  GETPOT ALSO READS IN THE CATCHMENT        GET00040
C  NUMBER OF THE "TRASHCAN".                                            GET00050
C                                                                       GET00060
C                                                                       GET00070
      IMPLICIT INTEGER*2 (I-N)                                          GET00080
      INTEGER LUNUIN, LUNPRT, LUNPIN, LUNNPS, LUNEXM, ITEMP             GET00090
      COMMON /POTEN/ PFACIN, IPFLAG                                     GET00100
      COMMON /UIO/ LUNUIN                                               GET00110
      COMMON /IO/ LUNPRT, LUNPIN, LUNNPS, LUNEXM                        GET00120
      DIMENSION PFACIN(2,12,5), LUNNPS(20)                              GET00130
      READ( LUNUIN, 10 ) IPFLAG                                         GET00140
   10 FORMAT( I1 )                                                      GET00150
      IF( IPFLAG .NE. 1 ) GOTO 100                                      GET00160
      READ( LUNUIN, 20 ) PFACIN                                         GET00170
   20 FORMAT( 120F10.6 )                                                GET00180
      RETURN                                                            GET00190
C                                                                       GET00200
  100 DO 110 I = 1, 2                                                   GET00210
         DO 120 J = 1, 12                                               GET00220
            DO 130 K = 1,5                                              GET00230
               PFACIN(I,J,K) = 0.0                                      GET00240
  130       CONTINUE                                                    GET00250
  120    CONTINUE                                                       GET00260
  110 CONTINUE                                                          GET00270
      RETURN                                                            GET00280
      END                                                               INI00010
      SUBROUTINE INIT                                                   INI00020
C                                                                       INI00030
C  INIT INITIALIZES SOME KEY FLAGS, AND CALLS GETPOT TO GET THE MONTHLY INI00040
C  BACKGROUND POTENCY FACTORS.                                          INI00050
C
```

```
C     IMPLICIT INTEGER*2 (I-N)                                          INI00060
      COMMON /FLAGS/ ITYPE, IFLAG, IYEAR, IMONTH, NDAYS, IYRCT          INI00070
      IFLAG = 0                                                         INI00080
      IYEAR = -1                                                        INI00090
      IMONTH = -1                                                       INI00100
      NDAYS = 0                                                         INI00110
      IYRCT = 1                                                         INI00120
      CALL GETPOT                                                       INI00130
      RETURN                                                            INI00140
      END                                                               INI00150
      SUBROUTINE MONCLR                                                 MON00010
C                                                                       MON00020
C  MONCLR CLEARS THE MONTHLY FLUX SUM ARRAYS FROM WHICH MONTHLY POTENCY MON00030
C  FACTORS ARE DETERMINED.                                              MON00040
C                                                                       MON00050
C     IMPLICIT INTEGER*2 (I-N)                                          MON00060
      COMMON /MATS/ AFLUX, PFAC, DOTRO1, RFLUX, DOTRO2, FLXSUM, TSPMAT  MON00070
      DIMENSION AFLUX(12,31), PFAC(2), DOTRO1(10), RFLUX(24,3),         MON00080
     *      FOTRO2(10), FLXSUM(3,20), TSPMAT(20,12,31)                  MON00090
      DO 10 I = 1, 3                                                    MON00100
         DO 20 J = 1, 20                                                MON00110
               FLXSUM(I,J) = 0.0                                        MON00120
  20     CONTINUE                                                       MON00130
   10 CONTINUE                                                          MON00140
      RETURN                                                            MON00150
      END                                                               MON00160
      SUBROUTINE NEWMON                                                 NEW00010
C                                                                       NEW00020
C     IMPLICIT INTEGER*2 (I-N)                                          NEW00030
      COMMON /FLAGS/ ITYPE, IFLAG, IYEAR, IMONTH, NDAYS, IYRCT          NEW00040
      IF( IMONTH .LT. 0 ) GOTO 500                                      NEW00050
C                                                                       NEW00060
C  CALCULATE AND PRINT MONTHLY POTENCIES.                               NEW00070
C                                                                       NEW00080
         CALL PRTPOT                                                    NEW00090
C                                                                       NEW00100
C  CLEAR MONTHLY ARRAYS.                                                NEW00110
C                                                                       NEW00120
  500 CALL MONCLR                                                       NEW00130
C                                                                       NEW00140
      RETURN                                                            NEW00150
      END                                                               NEW00160
      SUBROUTINE NEWYR                                                  NEW00010
C     IMPLICIT INTEGER*2 (I-N)                                          NEW00020
      COMMON /FLAGS/ ITYPE, IFLAG, IYEAR, IMONTH, NDAYS, IYRCT          NEW00030
      IF( IYEAR .LT. 0 ) GOTO 500                                       NEW00040
C                                                                       NEW00050
C  CALCULATE AND PRINT MONTHLY POTENCIES.                               NEW00060
C                                                                       NEW00070
         CALL PRTPOT                                                    NEW00080
         IYRCT = IYRCT + 1                                              NEW00090
C                                                                       NEW00100
C  CALCULATE AND PRINT ANNUAL TSP FLUX.                                 NEW00110
C                                                                       NEW00120
         CALL PRTTSP                                                    NEW00130
C                                                                       NEW00140
C  CLEAR MONTHLY AND ANNUAL ARRAYS.                                     NEW00150
```

```
C
  500 CALL MONCLR                                                       NEW00160
      CALL YRCLR                                                        NEW00170
C                                                                       NEW00180
      RETURN                                                            NEW00190
      END                                                               NEW00200
      FUNCTION NUMDAY( IMON, IYR )                                      NEW00210
C                                                                       NUM00010
C  GIVEN THE MONTH (IMON: 1-12) AND YEAR (IYR: A TWO OR MORE DIGIT      NUM00020
C  NUMBER FOR THE YEAR), THE NUMDAY FUNCTION RETURNS THE NUMBER OF DAYS NUM00030
C  IN THE MONTH.                                                        NUM00040
C                                                                       NUM00050
C     IMPLICIT INTEGER*2 (I-N)                                          NUM00060
      INTEGER ITEMP                                                     NUM00070
      ITEMP = IYR                                                       NUM00080
      IF( IMON .EQ. 1 .OR. IMON .EQ. 3 .OR. IMON .EQ. 5 .OR.            NUM00090
     *    IMON .EQ. 7 .OR. IMON .EQ. 8 .OR. IMON .EQ. 10 .OR.           NUM00100
     *    IMON .EQ. 12 ) GOTO 100                                       NUM00110
      NUMDAY = 28                                                       NUM00120
      IF( IMON .NE. 2 ) RETURN                                          NUM00130
      IF( MOD(ITEMP,4) .NE. 0 .OR. MOD(ITEMP,100) .EQ. 0 ) RETURN       NUM00140
      NUMDAY = 29                                                       NUM00150
      RETURN                                                            NUM00160
  100 NUMDAY = 31                                                       NUM00170
      RETURN                                                            NUM00180
      END                                                               NUM00190
      SUBROUTINE PRTPOT                                                 NUM00200
C                                                                       PRT00010
C  PRTPOT CALCULATES AND PRINTS MONTHLY POTENCY FACTORS BASED ON THE    PRT00020
C  MONTHLY TOTAL FLUX OF EACH POLLUTANT, AND BASED ON MONTHLY BACKGROUNDPRT00030
C  POTENCY FACTORS IF THEY ARE GIVEN (I.E., NON-ZERO).                  PRT00040
C                                                                       PRT00050
C     IMPLICIT INTEGER*2 (I-N)                                          PRT00060
      INTEGER LUNPRT, LUNPIN, LUNNPS, LUNEXM, ITEMP                     PRT00070
      COMMON /FLAGS/ ITYPE, IFLAG, IYEAR, IMONTH, NDAYS, IYRCT          PRT00080
      COMMON /IO/ LUNPRT, LUNPIN, LUNNPS, LUNEXM                        PRT00090
      COMMON /POLKAT/ NPOLLS, KATDEF                                    PRT00100
      COMMON /MATS/ AFLUX, PFAC, DOTRO1, RFLUX, DOTRO2, FLXSUM, TSPMAT   PRT00110
      COMMON /POTEN/ PFACIN, IPFLAG                                     PRT00120
      DIMENSION LUNNPS(20)                                              PRT00130
      DIMENSION KATDEF(20)                                              PRT00140
      DIMENSION AFLUX(12,31), PFAC(2), DOTRO1(10), RFLUX(24,3),         PRT00150
     *          DOTRO2(10), FLXSUM(3,20), TSPMAT(20,12,31)              PRT00160
      DIMENSION PFACIN(2,12,5)                                          PRT00170
      PFAC(1) = 0.0                                                     PRT00180
      PFAC(2) = 0.0                                                     PRT00190
      DO 10 I = 1, 20                                                   PRT00200
         IF( KATDEF(I) .NE. 1 ) GOTO 10                                 PRT00210
         IF( FLXSUM(1,I) .NE. 0.0 ) GOTO 5                              PRT00220
            WRITE(LUNPRT,1000) I, IYEAR, IMONTH                         PRT00230
            STOP                                                        PRT00240
5        ITOP = NPOLLS - 1                                              PRT00250
         DO 6 J = 1, ITOP                                               PRT00260
            PFAC(J) = FLXSUM(J+1,I) / FLXSUM(1,I)                       PRT00270
            IF( PFACIN(J,IMONTH,IYRCT) .EQ. 0.0 .OR.                    PRT00280
     *          IPFLAG .EQ. 0 ) GOTO 6                                  PRT00290
               PFAC(J) = (NDAYS*PFAC(J) + NUMDAY(IMONTH,IYEAR)          PRT00300
                                                                        PRT00310
     *                  * PFACIN(1,IMONTH,IYRCT))                     PRT00320
     *                  / (NDAYS + NUMDAY(IMONTH,IYEAR))              PRT00330
    6     CONTINUE
          ITEMP = LUNNPS(I)                                           PRT00340
          INDEX = 100 + IMONTH                                        PRT00350
C                                                                     PRT00360
C -------
C         CHANGE MADE TO OUTPUT 80 COLUMN RECORDS. RECORDS WERE
C         TO LONG WITHOUT THE USE OF AN OPEN STATEMENT.
C
C         D. CLINE, AERL, EPA - 3-AUG-81
C -------
C
          WRITE( ITEMP, 876 ) I, IYEAR, INDEX
          WRITE( ITEMP, 877 ) PFAC
C 876     FORMAT( I2, I4, I6, 31E11.4 )
  876     FORMAT( I2, I4, I6, 2H02 )                                  PRT00380
  877     FORMAT( 7E11.4)
   10 CONTINUE
      NDAYS = 0                                                       PRT00390
      RETURN                                                          PRT00400
 1000 FORMAT( ' ZERO TSP FLUX IN CATCHMENT ', I2, ', YEAR ', I4,      PRT00410
     *        ', MONTH ', I2, '.  PROGRAM HALTED.' )                  PRT00420
      END                                                             PRT00430
      SUBROUTINE PRTTSP                                               PRT00440
C                                                                     PRT00010
C  PRTTSP PRINTS DAILY TSP FLUX PER CATCHMENT FOR ONE YEAR.           PRT00020
C                                                                     PRT00030
C
      IMPLICIT INTEGER*2 (I-N)                                        PRT00040
      INTEGER LUNPRT, LUNPIN, LUNNPS, LUNEXM, ITEMP                   PRT00050
      COMMON /IO/ LUNPRT, LUNPIN, LUNNPS, LUNEXM                      PRT00060
      COMMON /FLAGS/ ITYPE, IFLAG, IYEAR, IMONTH, NDAYS, IYRCT        PRT00070
      COMMON /POLKAT/ NPOLLS, KATDEF                                  PRT00080
      COMMON /MATS/ AFLUX, PFAC, DOTRO1, RFLUX, DOTRO2, FLXSUM, TSPMAT PRT00090
      DIMENSION LUNNPS(20)                                            PRT00100
      DIMENSION KATDEF(20)                                            PRT00110
      DIMENSION AFLUX(12,31), PFAC(2), DOTRO1(10), RFLUX(24,3),       PRT00120
     *          DOTRO2(10), FLXSUM(3,20), TSPMAT(20,12,31)            PRT00130
      IZERO = 0                                                       PRT00140
      DO 20 I = 1, 20                                                 PRT00150
         IF( KATDEF(I) .NE. 1 ) GOTO 20                               PRT00160
         DO 16 J = 1, 12                                              PRT00170
              DO 14 K = 1, 31                                         PRT00180
                   AFLUX(J,K) = TSPMAT(I,J,K)                         PRT00190
   14         CONTINUE                                                PRT00200
   16 `   CONTINUE                                                    PRT00210
         ITEMP = LUNNPS(I)                                            PRT00220
C                                                                     PRT00230
C -------
C        CHANGE MADE TO OUTPUT 80 COLUMN RECORDS. RECORDS WERE TO
C        LONG WITHOUT THE USE OF AN OPEN STATEMENT.
C
C        D. CLINE, AERL, EPA - 3-AUG-81
C -------
C
         DO 888 J = 1, 12
              WRITE( ITEMP, 876 ) I, IYEAR, J                         PRT00240
```

```
                 WRITE( ITEMP, 877 ) ( AFLUX(J,K), K = 1,31 )
C 876            FORMAT( I2, I4, I6, 31E11.4 )                          PRT00260
  876            FORMAT( I2, I4, I6, 2H31)
  877            FORMAT(7E11.4)
  888       CONTINUE                                                    PRT00270
   20 CONTINUE                                                          PRT00280
      RETURN                                                            PRT00290
      END                                                               PRT00300
      SUBROUTINE REC1                                                   REC00010
C                                                                       REC00020
C  REC1 GETS THE NUMBER OF POLLUTANTS AND THE CATCHMENT DEFINITIONS     REC00030
C  FROM THE FIRST RECORD TYPE 1, AND IGNORES ALL SUCCEEDING RECORD      REC00040
C  TYPE 1'S.                                                            REC00050
C                                                                       REC00060
C     IMPLICIT INTEGER*2 (I-N)                                          REC00070
      INTEGER LUNPRT, LUNPIN, LUNNPS, LUNEXM, ITEMP                     REC00080
      COMMON /IO/ LUNPRT, LUNPIN, LUNNPS, LUNEXM                        REC00090
      COMMON /FLAGS/ ITYPE, IFLAG, IYEAR, IMONTH, NDAYS, IYRCT          REC00100
      COMMON /POLKAT/ NPOLLS, KATDEF                                    REC00110
      DIMENSION KATDEF(20), LUNNPS(10)                                  REC00120
      IF( IFLAG .EQ. 0 ) GOTO 10                                        REC00130
         READ( LUNPIN )                                                 REC00140
         RETURN                                                         REC00150
C                                                                       REC00160
   10 IFLAG = 1                                                         REC00170
      READ( LUNPIN ) NPOLLS, KATDEF                                     REC00180
C                                                                       REC00190
      IF( NPOLLS .GE. 2 .AND. NPOLLS .LE. 3 ) GOTO 20                   REC00200
         WRITE( LUNPRT, 100 ) NPOLLS                                    REC00210
      STOP                                                              REC00220
C                                                                       REC00230
   20 IDEF = 0                                                          REC00240
      DO 30 I = 1, 20                                                   REC00250
         IF( KATDEF(I) .GE. 0 .AND. KATDEF(I) .LE. 2 ) GOTO 40          REC00260
              WRITE( LUNPRT, 110 ) I, KATDEF(I)                         REC00270
              STOP                                                      REC00280
   40    IF( KATDEF(I) .GT. 0 ) IDEF = 1                                REC00290
   30 CONTINUE                                                          REC00300
      IF( IDEF .NE. 0 ) RETURN                                          REC00310
      WRITE( LUNPRT, 120 )                                              REC00320
      STOP                                                              REC00330
  100 FORMAT( ' TOO FEW OR TOO MANY POLLUTANTS, ', I3, ', IN RECORD ',  REC00340
     *              'TYPE 1.  PROGRAM HALTED.' )                        REC00350
  110 FORMAT( ' CATCHMENT DEFINITION NUMBER ', I2, ' IS ', I2, ', NOT ',REC00360
     *              'BETWEEN 0 AND 2.  PROGRAM HALTED.' )               REC00370
  120 FORMAT( ' NO DEFINED CATCHMENTS (ALL ZEROES IN KATDEF ARRAY).  ', REC00380
     *              'PROGRAM HALTED.' )                                 REC00390
      END                                                               REC00400
      SUBROUTINE REC2                                                   REC00010
C     IMPLICIT INTEGER*2 (I-N)                                          REC00020
      INTEGER LUNPRT, LUNPIN, LUNNPS, LUNEXM, ITEMP                     REC00030
      COMMON /IO/ LUNPRT, LUNPIN, LUNNPS, LUNEXM                        REC00040
      COMMON /FLAGS/ ITYPE, IFLAG, IYEAR, IMONTH, NDAYS, IYRCT          REC00050
      COMMON /POLKAT/ NPOLLS, KATDEF                                    REC00060
      COMMON /MATS/ AFLUX, PFAC, DOTRO1, RFLUX, DOTRO2, FLXSUM, TSPMAT  REC00070
      DIMENSION LUNNPS(20)                                              REC00080
      DIMENSION KATDEF(20), DRYFLX(5,20)                                REC00090
      DIMENSION AFLUX(12,31), PFAC(2), DOTRO1(10), RFLUX(24,3),          REC00100
     *          DOTRO2(10), FLXSUM(3,20), TSPMAT(20,12,31)                REC00110
      IF( IFLAG .GT. 0 ) GOTO 10                                          REC00120
         WRITE( LUNPRT, 100 )                                             REC00130
         STOP                                                             REC00140
C                                                                         REC00150
   10 READ( LUNPIN ) NYEAR, NMONTH, IDAYNO, IRON,                         REC00160
     *              ( IM, ( DRYFLX(M,N), N = 1, 20 ), M = 1, NPOLLS )     REC00170
      IF( IYEAR .NE. NYEAR ) CALL NEWYR                                   REC00180
      IF( IYEAR .EQ. NYEAR .AND. IMONTH .NE. NMONTH ) CALL NEWMON         REC00190
      IYEAR = NYEAR                                                       REC00200
      IMONTH = NMONTH                                                     REC00210
      NDAYS = NDAYS + 1                                                   REC00220
      CALL DAYMON( IDAYNO, IYEAR, JMONTH, JDAY )                          REC00230
      IF( IMONTH .EQ. JMONTH ) GOTO 20                                    REC00240
         WRITE( LUNPRT, 101 ) IMONTH, IDAYNO                              REC00250
         STOP                                                             REC00260
C                                                                         REC00270
   20 DO 30 I = 1, 20                                                     REC00280
         IF( KATDEF(I) .NE. 1 ) GOTO 30                                   REC00290
         TSPMAT(I,JMONTH,JDAY) = DRYFLX(1,I) * 4.463E-9                   REC00300
   30 CONTINUE                                                            REC00310
C                                                                         REC00320
      DO 40 I = 1, 20                                                     REC00330
         IF( KATDEF(I) .NE. 1 ) GOTO 40                                   REC00340
         DO 35 J = 1, NPOLLS                                              REC00350
              FLXSUM(J,I) = FLXSUM(J,I) + DRYFLX(J,I)                     REC00360
   35    CONTINUE                                                         REC00370
   40 CONTINUE                                                            REC00380
C                                                                         REC00390
C                                                                         REC00400
      DO 50 I = 2, NPOLLS                                                 REC00410
         J = 0                                                            REC00420
         DO 45 K = 1, 20                                                  REC00430
              IF( KATDEF(K) .NE. 2 ) GOTO 45                              REC00440
                   J = J + 1                                              REC00450
                   DOTRO1(J) = DRYFLX(I,K) * 1.0E-9                       REC00460
   45    CONTINUE                                                         REC00470
         IF( J .EQ. 0 ) GOTO 60                                           REC00480
         IF( J .EQ. 10 ) GOTO 48                                          REC00490
         M = J + 1                                                        REC00500
         DO 47 K = M, 10                                                  REC00510
              DOTRO1(K) = 0.0                                             REC00520
   47    CONTINUE                                                         REC00530
         IRFLAG = 0                                                       REC00540
         IF( IRON .GT. 0 ) IRFLAG = 1                                     REC00550
   48    WRITE( LUNEXM, 1000 ) I, DOTRO1, IRFLAG                          REC00560
 1000    FORMAT( I1, 10E11.4, I1 )                                        REC00570
   50 CONTINUE                                                            REC00580
   60 RETURN                                                              REC00590
  100 FORMAT( ' RECORD TYPE 2 MUST NOT APPEAR BEFORE THE FIRST ',         REC00600
     *        'RECORD TYPE 1.  PROGRAM HALTED.')                          REC00610
  101 FORMAT( ' MONTH NUMBER, ', I2, ', NOT COMPATABLE WITH DAY ',        REC00620
     *        'NUMBER, ', I4, '.  PROGRAM HALTED.' )                      REC00630
      END                                                                 REC00640
      SUBROUTINE REC3                                                     REC00010
C     IMPLICIT INTEGER*2 (I-N)                                            REC00020
```

```
      INTEGER*4 INDEX,TEMP1,TEMP2                                       REC00030
      INTEGER LUNPRT, LUNPIN, LUNNPS, LUNEXM, ITEMP                     REC00040
      COMMON /IO/ LUNPRT, LUNPIN, LUNNPS, LUNEXM                        REC00050
      COMMON /FLAGS/ ITYPE, IFLAG, IYEAR, IMONTH, NDAYS, IYRCT          REC00060
      COMMON /POLKAT/ NPOLLS, KATDEF                                    REC00070
      COMMON /MATS/ AFLUX, PFAC, DOTRO1, RFLUX, DOTRO2, FLXSUM, TSPMAT  REC00080
      COMMON /WTMATS/ WETFLX, EXMWET                                    REC00090
      DIMENSION LUNNPS(20)                                              REC00100
      DIMENSION KATDEF(20), DRYFLX(5,20)                                REC00110
      DIMENSION AFLUX(12,31), PFAC(2), DOTRO1(10), RFLUX(24,3),         REC00120
     *          DOTRO2(10), FLXSUM(3,20), TSPMAT(20,12,31)              REC00130
      DIMENSION WETFLX(3,24), EXMWET(20,3)                              REC00140
      IF( IFLAG .GT. 0 ) GOTO 10                                        REC00150
         WRITE( LUNPRT, 100 )                                           REC00160
         STOP                                                           REC00170
   10 IDAYNO = -1                                                       REC00180
C                                                                       REC00190
      DO 4 I = 1, 24                                                    REC00200
         DO 3 J = 1, 3                                                  REC00210
               RFLUX(I,J) = 0.0                                         REC00220
    3    CONTINUE                                                       REC00230
    4 CONTINUE                                                          REC00240
C                                                                       REC00250
      ICT = 0                                                           REC00260
      DO 20 I = 1, 20                                                   REC00270
         IF( KATDEF(I) .GT. 0 ) ICT = ICT + 1                           REC00280
   20 CONTINUE                                                          REC00290
C                                                                       REC00300
C                                                                       REC00310
   30 READ( LUNPIN ) NYEAR, NMONTH, NDAYNO, ICATCH,                     REC00320
     *          ( IM, (WETFLX(M,KHR), KHR = 1, 24 ), M = 1, NPOLLS )    REC00330
      IF( IDAYNO .EQ. -1 ) IDAYNO = NDAYNO                              REC00340
      IF( NDAYNO .EQ. IDAYNO ) GOTO 40                                  REC00350
         WRITE( LUNPRT, 1000 )                                          REC00360
         STOP                                                           REC00370
   40 IF( KATDEF(ICATCH) .NE. 1 ) GOTO 101                              REC00380
C                                                                       REC00390
C                                                                       REC00400
      DO 50 I = 1, 24                                                   REC00410
         DO 45 J = 1, NPOLLS                                            REC00420
               RFLUX(I,J) = WETFLX(J,I) * 4.463E-9                      REC00430
   45    CONTINUE                                                       REC00440
   50 CONTINUE                                                          REC00450
      ITEMP = LUNNPS(ICATCH)
C
C-----
C     CODE ADDED TO COMPUTE THE I*4 VALUE (PLUS, MOVED IT OUT OF THE LOOP)
C-----
C
      TEMP1 = 1000
      TEMP2 = IDAYNO
      TEMP2 = TEMP1 * TEMP2                                             REC00460
      DO 77 I = 1, NPOLLS
         TEMP1 = I
         INDEX = TEMP2 + TEMP1                                          REC00470
C        INDEX = 1000 * IDAYNO + I
         WRITE( ITEMP, 74 ) ICATCH,NYEAR,INDEX
```

```
          WRITE( ITEMP, 744 ) (RFLUX(J,I),J=1,24)
C  74     FORMAT( I2, I4, I6, 31E11.4 )
C                                                                        REC00490
C-------
C         FORMAT CHANGE MADE TO OUTPUT 80-COLUMN RECORDS
C-------
C
74        FORMAT( I2, I4, I6, 2H24)
744       FORMAT(7E11.4)
   77 CONTINUE
      ICT = ICT - 1                                                      REC00500
      IF( ICT .EQ. 0 ) GOTO 200                                          REC00510
      READ( LUNPIN ) ITYPE                                               REC00520
      IF( ITYPE .EQ. 3 ) GOTO 30                                         REC00530
         WRITE( LUNPRT, 5000 )                                           REC00540
         STOP                                                            REC00550
C                                                                        REC00560
C                                                                        REC00570
  101 IF( KATDEF(ICATCH) .EQ. 2 ) GOTO 110                               REC00580
         WRITE( LUNPRT, 5100 )                                           REC00590
         STOP                                                            REC00600
  110 DO 120 I = 1, NPOLLS                                               REC00610
         EXMWET(ICATCH,I) = 0.0                                          REC00620
         DO 115 J = 1, 24                                                REC00630
              EXMWET(ICATCH,I) = EXMWET(ICATCH,I) + WETFLX(I,J)          REC00640
  115    CONTINUE                                                        REC00650
         EXMWET(ICATCH,I) = EXMWET(ICATCH,I) * 1.0E-9                    REC00660
  120 CONTINUE                                                           REC00670
      ICT = ICT - 1                                                      REC00680
      IF( ICT .EQ. 0 ) GOTO 200                                          REC00690
      READ( LUNPIN ) ITYPE                                               REC00700
      IF( ITYPE .EQ. 3 ) GOTO 30                                         REC00710
         WRITE( LUNPRT, 5000 )                                           REC00720
         STOP                                                            REC00730
C                                                                        REC00740
C                                                                        REC00750
  200 DO 250 I = 2, NPOLLS                                               REC00760
         J = 0                                                           REC00770
         DO 210 K = 1, 20                                                REC00780
              IF( KATDEF(K) .NE. 2 ) GOTO 210                            REC00790
                   J = J + 1                                             REC00800
                   DOTRO2(J) = EXMWET(K,I)                               REC00810
  210    CONTINUE                                                        REC00820
         IF( J .EQ. 0 ) GOTO 300                                         REC00830
         IF( J .EQ. 10 ) GOTO 230                                        REC00840
         M = J + 1                                                       REC00850
         DO 220 K = M, 10                                                REC00860
              DOTRO2(K) = 0.0                                            REC00870
  220    CONTINUE                                                        REC00880
  230    WRITE( LUNEXM, 1000 ) I, DOTRO2                                 REC00890
 1000    FORMAT( I1, 10E11.4, I1 )                                       REC00900
  250 CONTINUE                                                           REC00910
  300 RETURN                                                             REC00920
  100 FORMAT( ' RECORD TYPE 3 MUST NOT APPEAR BEFORE THE FIRST ',        REC00930
     *       'RECORD TYPE 1.  PROGRAM HALTED.')                          REC00940
 5000 FORMAT( ' NOT ENOUGH WET DEPOSITION RECORDS.  PROGRAM HALTED.')    REC00950
 5100 FORMAT( ' BAD CATCHMENT NUMBER IN RECORD TYPE 3.  PROG HALTED.')   REC00960
                                                                         REC00970
      END                                                               REC00980
      SUBROUTINE YRCLR                                                  YRC00010
C     IMPLICIT INTEGER*2 (I-N)                                          YRC00020
      COMMON /MATS/ AFLUX, PFAC, DOTRO1, RFLUX, DOTRO2, FLXSUM, TSPMAT  YRC00030
      DIMENSION AFLUX(12,31), PFAC(2), DOTRO1(10), RFLUX(24,3),         YRC00040
     #     DOTRO2(10), FLXSUM(3,20), TSPMAT(20,12,31)                   YRC00050
      DO 10 I = 1, 20                                                   YRC00060
         DO 20 J = 1, 12                                                YRC00070
              DO 30 K = 1, 31                                           YRC00080
                   TSPMAT(I,J,K) = 0.0                                  YRC00090
30            CONTINUE                                                  YRC00100
20       CONTINUE                                                       YRC00110
10    CONTINUE                                                          YRC00120
      RETURN                                                            YRC00130
      END                                                               YRC00140
```

NPSDEP

```
C       *************************************************************
C       *                                                           *
C       *          NONPOINT SOURCE POLLUTANT LOADING (NPS) MODEL    *
C       *                                                           *
C       *************************************************************
C
C                           VERSION DATED: NOV. 15, 1976
C
C                               DEVELOPED BY:  HYDROCOMP, INCORPORATED
C                                              1502 PAGE MILL ROAD
C                                              PALO ALTO, CA.  94304
C                                                415-493-5522
C
C                                        FOR:  U.S. ENVIRONMENTAL
C                                                PROTECTION AGENCY
C                                              OFFICE OF RESEARCH
C                                               AND DEVELOPMENT
C                                              ENVIRONMENTAL
C                                               RESEARCH LABORATORY
C                                              ATHENS, GA.  30601
C                                                404-546-3581
C____________________________________________________________________
C
C              ALTERED MARCH 1981 TO MAKE COMPATIBLE WITH
C              AIR QUALITY MODEL 'DIDOT' AND WATER QUALITY MODEL
C              'EXAMS.'
C
C
C                                         ROBERT MCARTHUR
C                                         ARTHUR D. LITTLE, INC.
C                                         ACORN PARK
C                                         CAMBRIDGE, MASS.
C                                         (617) 864-5770
C____________________________________________________________________
C
C
C
C                             NPS - MAIN PROGRAM
      IMPLICIT  REAL(L)
      LOGICAL LAST, PREV
C
      LOGICAL LEAPYR
C**********
      LOGICAL IDBUG,LDBUG,DOT,EXAMS
C**********
      INTEGER  BGNDAY, BGNMON, BGNYR, ENDDAY, ENDMON, ENDYR,
     1         DYSTRT, DYEND, YEAR, DAY, H, HYCAL, TIME, PINT, PRINTR,
     2         YR, CN, TF, DA, DY, UNIT, SNOW, LMTS, RECOUT, SFLAG,
     3         TIMTIL, TILDAY, DPM
C
      REAL  IRC, NN, NNI, KV, K24L, KK24, INFIL, INTER,
     1      IFS, ICS, K24EL, K3, NEPTOM, NEPTOT, IDNS, MPACK,
     2      JRER, KRER, JSER, KSER,KEIM,JEIM, MELEV, KUGI,
     3      K1, KK4, IRC4, MELRAM, MELRAY, IPACK, IMPKO,
     4      INFTOM, INFTOT, IMPK, MMPIN, METOPT,
     5      KGPLB,KGPHA,MGPERL
```

```
C
      REAL*8  WSNAME,RTYPE,UTYPE
C
      DIMENSION MNAM(24),RAD(24),TEMPX(24),WINDX(24),RAIN(96),
     1          IRAIN(96),IRAD(12,31),IEVAP(12,31),IWIND(12,31),
     2          ITEMP(12,31,2),GRAD(24),RADDIS(24),WINDIS(24),
     3          AR1OUT(28),AR2OUT(28),COVVEC(12),REPERM(5,12),TCF(12),
     4          TOTAL(24),VMIN(24),VMAX(24),SD(24),RANGE(24),AVER(24),
     5          REIMPM(5,12),ACUIM(5,12),PMPTAB(5,5,12),PMITAB(5,5,12),
     6          ACUPM(5,12),PMDTAB(5,5,12)
C*********
      REAL IACCP,IACCI
      DIMENSION IACCP(5,12,31),IACCI(5,12,31),POLEX(2),WETDEP(24,2)
      DIMENSION DOTTOT(12)
C*********
C
      COMMON /ALL/ RU,HYMIN,HYCAL,DPST,UNIT,TIMFAC,LZS,AREA,RESB,SFLAG,
     1             RESB1,ROSB,SRGX,INTF,RGX,RUZB,UZSB,PERCB,RIB,P3,TF,
     2             KGPLB,LAST,PREV,TEMPX,IHR,IHRR,PR,RUI,A,PA,GWF,NOSY,
     3             SRER(5),TS(5),LNDUSE(3,5),AR(5),QUALIN(3,5),NOSI,NOS,
     4             NOSIM,UFL,UTMP,UNT1(2,2),UNT2(2,2),UNT3(2,2),WHGT,
     5             WHT,DEPW,ROSBI,RESBI,RESBI1,ARUN,LMTS(5),IMPK(5),
     6             NLAND,NQUAL,STMCH(200,24),RECOUT(5),FLOUT,SCALEF(5),
     7             SNOW,PACK,IPACK,SGEN(5)
C
      COMMON /LAND/DAY,PRTM,IMIN,IX,TWBAL,SGW,GWS,KV,LIRC4,LKK4,ALTR(9),
     1             UZS,IZ,UZSN,LZSN,INFIL,INTER,SGW1,DEC,DECI,TIT(13),
     2             K24L,KK24,K24EL,EP,IFS,K3,EPXMI,RESS1,RESS,SCEP,IRC,
     3             SRGXT1,MMPIN,KGPHA,METOPT,CCFAC,SCEP1,SRGXT,RAIN,SRC,
     4             SCF,IDNS,F,DGM,WC,MPACK,EVAPSN,MELEV,TSNOW,PETMIN,
     5             DEWX,DEPTH,MONTH,TMIN,PETMAX,ELDIF,SDEN,WINDX,INFTOM,
     6             TSNBAL,ROBTOM,ROBTOT,RXB,ROITOM,ROITOT,YEAR,CUNIT(7),
     7             INFTOT,MNAM,RAD,SRCI,FORM(42)
C
      COMMON /QLS/ WSNAME(6),KRER,JRER,KSER,JSER,TEMPCF,COVMAT(5,12),
     1             KEIM,JEIM,NDSR,ARP(5),ARI(5),ACCP(5),ACCI(5),RPER(5),
     2             PMP(5,5),PMI(5,5),QSNOW,SNOWY,SEDTM,SEDTY,SEDTCA,
     3             ACPOLP(5,5),ACERSN(5),APOLP(5,5),AERSN(5),COVER(5),
     4             APOLI(5,5),ACEIM(5),AEIM(5),POLTM(5),POLTY(5),
     5             TEMPA,DOA,POLTCA(5),AERSNY(5),AEIMY(5),APOLPY(5,5),
     6             APOLIY(5,5),POLTC(5),PLTCAY(5),ACPOLI(5,5),RIMP(5),
     $             PMD(5,5),STEPS
C
      COMMON /LNDOUT/ ROSTOM,RINTOM,RITOM,RUTOM,BASTOM,RCHTOM,PRTOM,
     1                SUMSNM,PXSNM,MELRAM,RADMEM,CONMEM,CDRMEM,
     2                CRAINM,SGMM,SNEGMM,PACKOT,SEVAPM,EPTOM,NEPTOM,
     3                UZSOT,LZSOT,SGWOT,SCEPOT,RESSOT,SRGXTO,TWBALO,
     4                TSNBOL,ROSTOT,RINTOT,RITOT,RUTOT,BASTOT,RCHTOT,
     5                PRTOT,SUMSNY,PXSNY,MELRAY,RADMEY,CONMEY,CDRMEY,
     6                CRAINY,SGMY,SNEGMY,PACK1,SEVAPY,EPTOT,NEPTOT,
     7                UZSMT,LZSMT,SGWMT,SCEPT,RESST,SRGXTT,TWBLMT
C
      COMMON /INTM/ RTYPE(4,4),UTYPE(2),GRAD,RADDIS,WINDIS,ICS,OFS,
     1              TEMPAY,DOAY,NOSIY,INTRVL,WMUL,NN,L,SS,NNI,LI,SSI,
     2              RMUL,KUGI,SEDTCY,REPERV(12),REIMPV(12),ACUPV(12),
     3              ACUIV(12),PMPMAT(12,5),PMIMAT(12,5),PMPVEC(5),
     4              PMIVEC(5),ACUI,ACUP,REIMP,REPER,PRINTR,
     5              EPXM(12),TIMTIL(12),SRERTL(12),TILDAY(5,12),           81.1
     6              TILSED(5,12),DPM(12),TCF,PMDVEC(5),PMDMAT(12,5)        82.
C
C**********
      COMMON/PARM/IDBUG,LDBUG,DOT,EXAMS,ISHED
      COMMON /EXOUT/ WETTMP(5),TOTROF(5),SEDCON(5),POLCON(5,5)
C
      COMMON/UNITS/NIN,NMET,NOUT,KR,ILP,ILUDOT
      EQUIVALENCE (IACCP(1,1,1),IACCI(1,1,1))
C**********                                                                 83.
      EQUIVALENCE (ROSTOM,AR1OUT(1)),(TSNBOL,AR2OUT(1))                   100.
C                      NAMELIST INPUT VARIABLES                             101.
C                                                                           102.
      NAMELIST /ROPT/  HYCAL, HYMIN, NLAND, NQUAL, SNOW,IDBUG,LDBUG
C**********
      NAMELIST /DTYP/  UNIT, PINT, MNVAR, DOT, EXAMS, ISHED
C**********                                                                 104.
      NAMELIST /STRT/  BGNDAY, BGNMON, BGNYR                                105.
      NAMELIST /ENDD/  ENDDAY, ENDMON, ENDYR                                106.
      NAMELIST /LND1/  UZSN, LZSN, INFIL, INTER ,IRC, AREA                  107.
      NAMELIST /LND2/  NN, L, SS, NNI, LI, SSI                              108.
      NAMELIST /LND3/  K1, PETMUL, K3 ,EPXM, K24L, KK24                     109.
      NAMELIST /LND4/  UZS, LZS, SGW                                        110.
      NAMELIST /SNO1/  RADCON, CCFAC, EVAPSN                                111.
      NAMELIST /SNO2/  MELEV, ELDIF, TSNOW                                  112.
      NAMELIST /SNO3/  MPACK, DGM, WC, IDNS                                 113.
      NAMELIST /SNO4/  SCF, WMUL, RMUL, F, KUGI                             114.
      NAMELIST /SNO5/  PACK, DEPTH                                          115.
      NAMELIST /WSCH/  ARFRAC ,IMPKO, COVVEC, TIMTIL, SRERTL
      NAMELIST /YPTM/  PMPVEC, PMIVEC,PMDVEC
      NAMELIST /MPTM/  PMPMAT,PMIMAT,PMDMAT                                 118.
      NAMELIST /WASH/  JRER, KRER, JSER ,KSER, JEIM, KEIM, TCF              119.
      NAMELIST /YACR/  ACUP,ACUI                                            120.
      NAMELIST /MACR/  ACUPV,ACUIV                                          121.
      NAMELIST /YRMR/  REPER, REIMP                                         122.
      NAMELIST /MRMR/  REPERV,REIMPV
      NAMELIST /INAC/  SRERI, TSI,SGENI                                     124.
C                                                                           125.
C          NAMELIST INPUT PARAMETER DESCRIPTION                             126.
C  HYCAL : INDICATES TYPE OF SIMULATION RUN                                 127.
C        =  1  HYDROLOGIC CALIBRATION                                       128.
C        =  2  SEDIMENTS AND QUALITY CALIBRATION                            129.
C        =  3  PRODUCTION RUN (PRINTER OUTPUT)
C        =  4  PRODUCTION RUN (PRINTER & FILE OUTPUT ON UNIT 3)
C  HYMIN : MINIMUM FLOW FOR OUTPUT DURING A TIME INTERVAL (CFS, CMS)       131.
C  UNIT  : ENGLISH(-1), METRIC(1)                                           132.
C  NLAND : NUMBER OF LAND TYPE USES IN THE WATERSHED                        133.
C  NQUAL : NUMBER OF QUALITY CONSTITUENTS SIMULATED                         134.
C  SNOW  : (0) SNOWMELT NOT PERFORMED, (1) SNOWMELT CALC'S PERFORMED       135.
C  MNVAR : MONTHLY VARIATION IN ACCUMULATION RATES, REMOVAL RATES,         136.
C          AND POTENCY FACTORS USED (1), OR NOT USED (0)                    137.
C
C**********
C  DOT   : LOGICAL FLAG TO INDICATE DIDOT INPUT (DAILY ACCUM. RATES)
C  EXAMS : LOGICAL FLAG TO INDICATE EXAMS OUTPUT (DAILY LOADINGS)
C  ISHED : WATERSHED NUMBER (USED IN CONJUNCTION WITH EXAMS INTERFACE)
C**********
C  PINT  : INPUT PRECIPITATION IN INTERVALS OF 15 MIN.(0), OR HOURLY (1) 138.
```

```
C  BGNDAY, BGNMON, BGNYR : DATE SIMULATION BEGINS
C  ENDDAY, ENDMON, ENDYR : DATE SIMULATION ENDS
C  UZSN   : NOMIMAL UPPER ZONE STORAGE (IN, MM)
C  LZSN   : NOMINAL LOWER ZONE STORAGE (IN, MM)
C  INFIL  : INFILTRATION RATE (IN/HR, MM/HR)
C  INTER  : INTERFLOW PARAMETER, ALTERS RUNOFF TIMING
C  IRC    : INTERFLOW RECESSION RATE
C  AREA   : WATERSHED AREA IN ACRES
C  NN     : MANNING'S N FOR OVERLAND PERVIOUS FLOW
C  NNI    : MANNING'S N FOR OVERLAND IMPERVIOUS FLOW
C  L      : LENGTH OF OVERLAND PERVIOUS FLOW TO CHANNEL (FT, M)
C  LI     : LENGTH OF OVERLAND IMPERVIOUS FLOW TO CHANNEL (FT, M)
C  SS     : AVERAGE OVERLAND PERVIOUS FLOW SLOPE
C  SSI    : AVERAGE OVERLAND IMPERVIOUS FLOW SLOPE
C  K1     : RATIO OF SPATIAL AVERAGE RAINFALL TO GAGE RAINFALL
C  K3     : INDEX TO ACTUAL EVAPORATION
C  PETMUL: POTENTIAL EVAPOTRANSPIRATION MULTIPLICATION FACTOR
C  EPXM   : INTERCEPTION STORAGE, 12 MONTHLY VALUES (IN,MM)
C  K24L   : FRACTION OF GROUNDWATER RECHARGE PERCOLATING TO DEEP
C           GROUNDWATER
C  KK24   : GROUNDWATER RECESSION RATE
C  UZS    : INITIAL UPPER ZONE STORAGE (IN, MM)
C  LZS    : INITIAL LOWER ZONE STORAGE (IN, MM)
C  SGW    : INITIAL GROUNDWATER STORAGE (IN, MM)
C  RADCON: CORRECTION FACTOR FOR RADIATION
C  CCFAC  : CORRECTION FACTOR FOR CONDENSATION AND CONVECTION
C  SCF    : SNOW CORRECTION FACTOR FOR RAINGAGE CATCH DEFICIENCY
C  ELDIF  : ELEVATION DIFFERENCE FROM TEMP. STATION TO MEAN SEGMENT ELEVA
C           (1000 FT, KM)
C  IDNS   : DENSITY OF NEW SNOW AT 0 DEGREES F.
C  F      : FRACTION OF SEGMENT WITH COMPLETE FOREST COVER
C  DGM    : DAILY GROUNDMELT (IN/DAY, MM/DAY)
C  WC     : MAXIMUM WATER CONTENT OF SNOWPACK BY WEIGHT
C  MPACK  : ESTIMATED WATER EQUIVALENT OF SNOWPACK FOR COMPLETE COVERAGE
C  EVAPSN: CORRECTION FACTOR FOR SNOW EVAPORATION
C  MELEV  : MEAN ELEVATION OF WATERSHED (FT, M)
C  TSNOW  : TEMPERATURE BELOW WHICH SNOW FALLS (F, C)
C  PACK   : INITIAL WATER EQUIVALENT OF SNOWPACK (IN, MM)
C  DEPTH  : INITIAL DEPTH OF SNOWPACK (IN, MM)
C  ARFRAC: PERCENT OF A GIVEN LAND TYPE USE
C  IMPKO  : PERCENTAGE OF IMPERVIOUS AREA FOR A GIVEN LAND TYPE USE
C  COVVEC: MONTHLY COVER COEFF. FOR A GIVEN LAND TYPE USE
C  PMPVEC: POTENCY FACTORS FOR A GIVEN LAND TYPE - PERVIOUS AREAS
C  PMPMAT: MATRIX OF POTENCY FACTORS FOR MONTHLY VARIATIONS
C           - PERVIOUUS AREAS, 12 VALUES PER CONSTITUENT
C  PMDVEC: POTENCY FACTORS FOR DEPOSITED SEDIMENT(PERVIOUS)
C  PMDMAT: MONTHLY POTENCY FACTORS FOR DEPOSITED SEDIMENT
C  PMIVEC: POTENCY FACTORS FOR A GIVEN LAND TYPE - IMPERVIOUS AREAS
C  PMIMAT: MATRIX OF POTENCY FACTORS FOR MONTHLY VARIATIONS
C           - IMPERVIOUS AREAS, 12 VALUES PER CONSTITUENT
C  TCF    : TEMPERATURE CORRECTION FACTOR RELATING RUNOFF AND
C           AIR TEMPERATURES
C  JRER   : EXPONENT IN RAINDROP SOIL SPLASH EQUATION
C  KRER   : COEF. IN RAINDROP SOIL SPLASH EQUATION
C  JSER   : EXPONENT IN WASH OFF FUNCTION FOR PERVIOUS AREAS
C  KSER   : COEF. IN WASH OFF FUNCTION FOR PERVIOUS AREAS
C  JEIM   : EXPONENT IN WASH OFF FUNCTION FOR IMPERVIOUS AREAS
```

```
C  KEIM  : COEF. IN WASH OFF FUNCTION FOR IMPERVIOUS AREAS
C  ACUI  : ACCUMULATION RATES - IMPERVIOUS AREAS
C  ACUIV : MONTHLY ACCUMULATION RATES - IMPERVIOUS AREAS, 12 VALUES
C  ACUP  : ACCUMULATION RATES - PERVIOUS AREAS
C  ACUPV : MONTHLY ACCULUMATION RATES - PERVIOUS AREAS, 12 VALUES
C  REIMP : REMOVAL COEF. -IMPERVIOUS AREAS
C  REIMPV: MONTHLY REMOVAL COEF.- IMPERVIOUS AREAS, 12 VALUES
C  REPER : REMOVAL COEF. - PERVIOUS AREAS
C  REPERV: MONTHLY REMOVAL COEF.- PERVIOUS AREAS, 12 VALUES
C  SRERI : INITIAL DEPOSITED FINES AVAILABLE FOR TRANSPORT
C  TSI   : INITIAL AMOUNT OF SOLIDS AVAILABLE FOR TRANSPORT
C  SGENI : INITIAL GENERATED FINES AVAILABLE FOR TRANSPORT
C  NOTE  : INPUTS FOR SRERI,TSI,SGENI AND SRERTL ARE TONS/ACRE
C          OR KG/HA
C
      DATA DEPLBL/4HDEPD/,GENLBL/4HGEND/
      DATA BLANKS/4H    /,GMPERL/4HG/L /,MGPERL/4HMG/L/
C
      READ(KR,4520) (WSNAME(I),I=1,6)
 4520 FORMAT(3A8,/,3A8)
      READ(KR,ROPT)
      READ(KR,DTYP)
      READ(KR,STRT)
      READ(KR,ENDD)
      READ(KR,LND1)
      READ(KR,LND2)
      READ(KR,LND3)
      READ(KR,LND4)
      IF (SNOW .LT. 1) GO TO 20
      QSNOW=SNOWY
      READ(KR,SNO1)
      READ(KR,SNO2)
      READ(KR,SNO3)
      READ(KR,SNO4)
      READ(KR,SNO5)
   20 READ(KR,WASH)
      DO 30 J=1,NQUAL
   30 READ(KR,4060) (QUALIN (I,J),I=1,3),CUNIT(J)
      DO 100 II=1,NLAND
      READ(KR,4060) (LNDUSE(K,II),K=1,3)
      READ(KR,WSCH)
      AR(II)=ARFRAC
      IMPK(II)=IMPKO
      DO 40 IJ=1,12
   40 COVMAT(II,IJ)=COVVEC(IJ)
C
      DO 42 JJ=1,12
      TILDAY(II,JJ) = TIMTIL(JJ)
   42 TILSED(II,JJ) = SRERTL(JJ)
C
      IF (MNVAR.EQ.1) GO TO 60
C
C          READ INPUT DATA OF ACCUMULATION RATES, REMOVAL RATES, AND
C          POTENCY MATRICES WITHOUT MONTHLY VARIATION
C
      READ(KR,YPTM)
      READ(KR,YACR)
      READ(KR,YRMR)
      DO 50 IJ=1,NQUAL
      PMPTAB(IJ,II,BGNMON)=PMPVEC(IJ)
      PMDTAB(IJ,II,BGNMON)=PMDVEC(IJ)
   50 PMITAB(IJ,II,BGNMON)=PMIVEC(IJ)
      ACUPM(II,BGNMON)=ACUP
      ACUIM(II,BGNMON)=ACUI
      REPERM(II,BGNMON)=REPER
      REIMPM(II,BGNMON)=REIMP
      GO TO 90
C
C           READ INPUT DATA OF ACCUMULATION RATES, REMOVAL RATES, AND
C           POTENCY MATRICES WITH MONTHLY VARIATION
C
   60 READ(KR,MPTM)
      READ(KR,MACR)
      READ(KR,MRMR)
      DO 70 IJ=1,NQUAL
      DO 70 MN=1,12
      PMPTAB(IJ,II,MN)=PMPMAT(MN,IJ)
      PMDTAB(IJ,II,MN)=PMDMAT(MN,IJ)
   70 PMITAB(IJ,II,MN)=PMIMAT(MN,IJ)
      DO 80 MN=1,12
      ACUPM(II,MN)=ACUPV(MN)
      ACUIM(II,MN)=ACUIV(MN)
      REPERM(II,MN)=REPERV(MN)
   80 REIMPM(II,MN)=REIMPV(MN)
   90 CONTINUE
      READ(KR,INAC)
      SRER(II)=SRERI
      SGEN(II)=SGENI
      TS(II)=TSI
      IF(IDBUG) WRITE(ILP,6004) II
 6004 FORMAT(' *****NPS MAIN--READ LAND USE INFO FOR TYPE ',I2)
  100 CONTINUE
      IF (UNIT .EQ. -1) GO TO 120
      DEPW=UNT1(2,1)
      WHGT=UNT1(1,1)
      WHT=UNT2(1,1)
      UFL=UNT2(2,1)
      UTMP=UNT3(1,1)
      ARUN=UNT3(2,1)
      KUNT=1
      GO TO 130
  120 DEPW=UNT1(2,2)
      WHGT=UNT1(1,2)
      WHT=UNT2(1,2)
      UFL=UNT2(2,2)
      UTMP=UNT3(1,2)
      ARUN=UNT3(2,2)
      KUNT=2
C
C                 PRINTING OF TITLE PAGE AND INPUT PARAMETERS
C
  130 WRITE(ILP,4070)
      WRITE(ILP,4080) (WSNAME(I),I=1,6),ARUN,AREA
      WRITE(ILP,4090) ARUN,ARUN,ARUN
      ARPT=0.0
      ARIT=0.0
      DO 140 I=1,NLAND
      TEM=AREA*AR(I)
      ARP(I)=TEM*(1.-IMPK(I))
      ARPT=ARPT+ARP(I)
      ARI(I)=TEM*IMPK(I)
      ARIT=ARIT+ARI(I)
      AR(I)=AR(I)*100.
      PER=IMPK(I)*100.
      WRITE(ILP,4100) (LNDUSE(KK,I),KK=1,3),AR(I),TEM,ARP(I),ARI(I),PER
      AR(I)=TEM
  140 CONTINUE
      A=ARIT/AREA
      PA=1.0-A
C     COMPUTE IMPERVIOUS AND PERVIOUS AREA FRACTIONS
      WRITE(ILP,4110) A
      IF (ABS((ARIT+ARPT-AREA)/AREA).LE.0.001) GO TO 150
      WRITE(ILP,4120)
      GO TO 1600
C
C                    PRINTING OF SIMULATION CHARACTERISTICS
C
  150 IZ=BGNMON*2-1
      IX=IZ+1
      IP=ENDMON*2-1
      IQ=IP+1
      NQI=NQUAL+3
      IF (PINT.EQ.1) PRINTR=60
      WRITE(ILP,4130) (RTYPE(HYCAL,I),I=1,4),MNAM(IZ),MNAM(IX),BGNDAY,
     *                BGNYR,MNAM(IP),MNAM(IQ),ENDDAY,ENDYR,PRINTR,INTRVL,
     *                QSNOW,UTYPE(KUNT),UTYPE(KUNT),UFL,
     *                HYMIN,NQI,((QUALIN(I,J),I=1,3),J=1,NQUAL)
      WRITE(ILP,4140) INTER,IRC,INFIL,NN,L,SS,NNI,LI,SSI,K1,
     *                PETMUL,K3,K24L,KK24,UZSN,LZSN
      IF (SNOW.EQ.1) WRITE(ILP,4150) RADCON,CCFAC,EVAPSN,MELEV,
     *                               ELDIF,TSNOW,MPACK,DGM,WC,IDNS,SCF,
     *                               WMUL,RMUL,F,KUGI
      WRITE(ILP,4160) JRER,KRER,JSER,KSER,JEIM,KEIM
      WRITE(ILP,4162)
      DO 158 I=1,NLAND
  158 WRITE(ILP,4165) (LNDUSE(K,I),K=1,3),(TILDAY(I,J),J=1,12),
     *                (TILSED(I,J),J=1,12)
      IF (MNVAR.EQ.1) GO TO 200
C
C           PRINTING OF ACCUMULATION RATES,REMOVAL RATES,
C           AND POTENCY FACTORS WITHOUT MONTHLY VARIATION
C
      IF(DOT) GO TO 162
      WRITE(ILP,4010)
      DO 160 I=1,NLAND
  160 WRITE(ILP,4230)(LNDUSE(K,I),K=1,3),ACUPM(I,BGNMON),ACUIM(I,BGNMON)
  162 CONTINUE
      WRITE(ILP,4010)
      DO 170 I=1,NLAND
  170 WRITE(ILP,4240) (LNDUSE(KK,I),KK=1,3),REPERM(I,BGNMON),
     *                REIMPM(I,BGNMON)
```

```
      WRITE(ILP,4250) GENLBL,((LNDUSE(KK,I),KK=1,3),I=1,NLAND)
      DO 180 I=1,NQUAL
  180 WRITE(ILP,4260) (QUALIN(J,I),J=1,3),(PMPTAB(I,K,BGNMON),K=1,NLAND)
      WRITE(ILP,4250) DEPLBL,((LNDUSE(KK,I),KK=1,3),I=1,NLAND)
      DO 182 I=1,NQUAL
  182 WRITE(ILP,4260) (QUALIN(J,I),J=1,3),(PMDTAB(I,K,BGNMON),K=1,NLAND)
      WRITE(ILP,4270) ((LNDUSE(KK,I),KK=1,3),I=1,NLAND)
      DO 190 I=1,NQUAL
  190 WRITE(ILP,4260) (QUALIN(J,I),J=1,3),(PMITAB(I,K,BGNMON),K=1,NLAND)
C
C  PRINTING OF MONTHLY COVER FUNCTION, EPXM  AND TEMP CORRECTION FACTORS
C
  200 WRITE(ILP,4170) (MNAM(I),I=1,24,2),(TCF(I),I=1,12),
     *          (EPXM(II),II=1,12),
     *          (LNDUSE(KK,1),KK=1,3),(COVMAT(1,KK),KK=1,12)
      IF (NLAND.EQ.1) GO TO 220
      DO 210 I=2,NLAND
  210 WRITE(ILP,4180) (LNDUSE(KK,I),KK=1,3),(COVMAT(I,KK),KK=1,12)
  220 IF (MNVAR.EQ.0) GO TO 290
C
C          PRINTING OF ACCUMULATION RATES,REMOVAL RATES,
C          AND POTENCY FACTORS WITH MONTHLY VARIATION
C
      WRITE(ILP,4190)
      DO 230 I=1,NLAND
  230 WRITE(ILP,4180) (LNDUSE(KK,I),KK=1,3),(ACUPM(I,J),J=1,12)
      WRITE(ILP,4200)
      DO 240 I=1,NLAND
  240 WRITE(ILP,4180) (LNDUSE(KK,I),KK=1,3),(REPERM(I,J),J=1,12)
      DO 250 J=1,NQUAL
      WRITE(ILP,4210) (QUALIN(KK,J),KK=1,3)
      DO 250 I=1,NLAND
  250 WRITE(ILP,4180) (LNDUSE(KK,I),KK=1,3),(PMPTAB(J,I,K),K=1,12)
      WRITE(ILP,4220)
      IF(DOT) GO TO 262
      WRITE(ILP,4190)
      DO 260 I=1,NLAND
  260 WRITE(ILP,4180) (LNDUSE(KK,I),KK=1,3),(ACUIM(I,J),J=1,12)
  262 CONTINUE
      WRITE(ILP,4200)
      DO 270 I=1,NLAND
  270 WRITE(ILP,4180) (LNDUSE(KK,I),KK=1,3),(REIMPM(I,J),J=1,12)
      DO 280 J=1,NQUAL
      WRITE(ILP,4210) (QUALIN(KK,J),KK=1,3)
      DO 280 I=1,NLAND
  280 WRITE(ILP,4180) (LNDUSE(KK,I),KK=1,3),(PMITAB(J,I,K),K=1,12)
C
C          PRINTING OF INITIAL CONDITIONS
C
  290 WRITE(ILP,4280) UZS,LZS,SGW
      IF (SNOW.EQ.1) WRITE(ILP,4290) PACK,DEPTH
      WRITE(ILP,4300) (LNDUSE(KK,1),KK=1,3),TS(1),SRER(1),SGEN(1)
      IF (NLAND.EQ.1) GO TO 310
      DO 300 I=2,NLAND
  300 WRITE(ILP,4310) (LNDUSE(KK,I),KK=1,3),TS(I),SRER(I),SGEN(I)
  310 IF (UNIT.EQ.-1) GO TO 350
C
```

```
C  CONVERSION OF METRIC INPUT DATA TO ENGLISH UNITS
C
      HYMIN= HYMIN*35.3
      UZSN = UZSN/MMPIN
      LZSN = LZSN/MMPIN
      INFIL= INFIL/MMPIN
      L    = L*3.281
      LI   = LI*3.281
      UZS  = UZS/MMPIN
      LZS  = LZS/MMPIN
      SGW  = SGW/MMPIN
      ICS  = ICS/MMPIN
      OFS  = OFS/MMPIN
      IFS  = IFS/MMPIN
      DO 315 I=1,12
  315 EPXM(I) = EPXM(I)/MMPIN
      AREA = AREA*2.471
      DO 340 I=1,NLAND
      AR(I)=AR(I)*2.471
      ARP(I)=ARP(I)*2.471
      ARI(I)=ARI(I)*2.471
      SRER(I)=SRER(I)/2.24
      TS(I)=TS(I)/2.24
      DO 318 JJ=1,12
  318 TILSED(I,JJ) = TILSED(I,JJ)/2.24
      IF (MNVAR.GT.0) GO TO 320
      ACUPM(I,BGNMON)=ACUPM(I,BGNMON)*KGPHA
      ACUIM(I,BGNMON)=ACUIM(I,BGNMON)*KGPHA
      GO TO 340
  320 DO 330 J=1,12
      ACUPM(I,J)=ACUPM(I,J)*KGPHA
  330 ACUIM(I,J)=ACUIM(I,J)*KGPHA
  340 CONTINUE
      DO 345 I=7,37,6
  345 FORM(I)=ALTR(2)
      IF (SNOW.LT.1) GO TO 350
      ELDIF = ELDIF/0.3048
      DGM   = DGM/MMPIN
      MPACK=MPACK/MMPIN
      MELEV = MELEV/0.3048
      TSNOW = 1.8*TSNOW + 32.0
      PACK  = PACK/MMPIN
      DEPTH = DEPTH/MMPIN
C
C                    ADJUSTMENT OF CONSTANTS
C
  350 H = 60/INTRVL
      TIMFAC = INTRVL
      INTRVL = 24*H
C     H     :  TIME PERIODS PER HOUR
C     TIMFAC:  LENGTH OF PERIOD IN MINUTES
C     INTRVL:  NUMBER OF PERIODS IN DAY
      ARIT=0.0
      KRER=KRER*H**(JRER-1.0)
      KSER=KSER*H**(JSER-1.0)
      KEIM=KEIM*H**(JEIM-1.0)
      DO 355 I=1,NQUAL
      SCALEF(I)=1.0
      IF (CUNIT(I).EQ.BLANKS) CUNIT(I)=GMPERL
C     ASSUME G/L FOR CONCENTRATION IF NO INPUT UNITS
      IF (CUNIT(I).EQ.MGPERL) SCALEF(I)=1000.
355   CONTINUE
      IF (NQUAL.EQ.5) GO TO 357
      II=11+NQUAL*6
      DO 356 I=II,40
  356 FORM(I)=ALTR(1)
  357 I=NQUAL+4
      TIT(I)=ALTR(I)
      J=0
      DO 358 I=15,39,6
      J=J+1
      IF(SCALEF(J).LE.2.) GO TO 358
      FORM(I)=ALTR(3)
  358 CONTINUE
      IF(HYCAL.NE.4) GO TO 364
C     WRITE HEADER FOR OUTPUT FILE
      WRITE(NOUT,3600) NQUAL,((QUALIN(I,J),I=1,3),J=1,NQUAL)
3600  FORMAT('SEDIMENT',T15,I1,T21,5(3A4))
C     WRITE LABELS FOR QUALITY CONSTITUENTS
      SEDIF=1.0
      WRITE(NOUT,3610) SEDIF,(SCALEF(J),J=1,NQUAL)
3610  FORMAT(T21,6F8.3)
364   CONTINUE
C
C          CONVERT ACCUMULATION RATES INTO TONS/ACRE/DAY
C
      DO 380 I=1,NLAND
      IF (MNVAR.GT.0) GO TO 365
      ACUPM(I,BGNMON)=ACUPM(I,BGNMON)/2000.
      ACUIM(I,BGNMON)=ACUIM(I,BGNMON)/2000.
      GO TO 380
  365 DO 370 J=1,12
      ACUIM(I,J)=ACUIM(I,J)/2000.
      ACUPM(I,J)=ACUPM(I,J)/2000.
  370 CONTINUE
  380 CONTINUE
        IRC4=IRC**(1.0/96.0)
        LIRC4=1.0-IRC4
        KK4=KK24**(1.0/96.0)
        LKK4= 1.0 - KK4
C
      IF ((24.*60./TIMFAC) .GT. 100.)  GO TO 390
      GO TO 400
  390 LIRC4 = LIRC4/3.0
      LKK4 = LKK4/3.0
  400 DEC= 0.00982*((NN*L/SQRT(SS))**0.6)
      SRC= 1020.*SQRT(SS)/(NN*L)
      DECI= 0.00982*((NNI*LI/SQRT(SSI))**0.6)
      SRCI= 1020.*SQRT(SSI)/(NNI*LI)
C                                   INITIALIZE IPACK
      IPACK=0.1*MPACK
        UZSB = UZS
        RESB = OFS
        SRGX = IFS
```

```
C
C                    INITIALIZE JCOUNT - DAY OF YEAR
C
      JCOUNT = BGNDAY - 1
      IF (BGNMON.EQ.1)  GO TO 404
      JJ = BGNMON - 1
      DO 402 I=1,JJ
  402 JCOUNT = JCOUNT + DPM(I)
  404 CONTINUE
      IF (MOD(BGNYR,4).EQ.0 .AND. BGNMON.GT.2) JCOUNT=JCOUNT+1
      RESS1 = OFS
      RESS = OFS
      SCEP = ICS
      SCEP1 = ICS
      SRGXT = IFS
      SRGXT1 = IFS
      SGW1 = SGW
C
C                         PROGRAM EXECUTION
C                                                BEGIN YEARLY LOOP
      DO 1590  YEAR=BGNYR,ENDYR
      LEAPYR=MOD(YEAR,4).EQ.0.AND.MOD(YEAR,100).NE.0
      IF(IDBUG) WRITE(ILP,6002) YEAR,LEAPYR
 6002 FORMAT(' *****STARTING YEAR LOOP--YEAR=',I4,' LEAPYR=',L1)
        MNSTRT = 1
        MNEND = 12
        IF (YEAR .EQ. BGNYR)  MNSTRT = BGNMON
        IF (YEAR .EQ. ENDYR)  MNEND = ENDMON
C
C
C  EVAP, TEMP(MAX-MIN), RAD, AND WIND  DATA INPUT
C
       DO  410  DA = 1,31
  410   READ(NMET,4050) (IEVAP(MN,DA), MN =1,12)
        DO  420 DA = 1,31
  420   READ(NMET,4040) ((ITEMP(MN,DA,IT), IT=1,2),MN=1,12)
      IF(YEAR.EQ.BGNYR) MAXPRV=ITEMP(1,1,1)
C     INITIALIZE PREVIOUS DAY'S TEMPERATURE AT FIRST DAY'S MAX
C
C**********
      IF(.NOT.DOT) GO TO 429
C     HERE TO READ DAILY ACCUMULATION RATES
      DO 422 JMON=1,12
      READ (ILUDOT,423) (IACCP(1,JMON,K),K=1,31)
      DO 422 KDAY=1,31
      DO 422 JLAND=2,NLAND
  422 IACCP(JLAND,JMON,KDAY)=IACCP(1,JMON,KDAY)
      IF (IDBUG) WRITE (ILP,424) ((IACCP(1,I,J),I=1,12),J=1,31)
  423 FORMAT (12X,31E11.4)
  424 FORMAT ('1','DIDOT ACCUM. RATES',///,31(' ',12E11.4,/))
C
      DO 425 MONTHO=MNSTRT,MNEND
      READ (ILUDOT,427) POTDOT
      DO 425 IO=1,NLAND
      PMITAB(1,IO,MONTHO)=POTDOT*100.
  425 PMDTAB(1,IO,MONTHO)=POTDOT*100.
      IF (IDBUG) WRITE (ILP,426) (MON,PMITAB(1,1,MON),MON=MNSTRT,MNEND)
```

```
C
  426 FORMAT (//,' DIDOT POTENCY BY MONTH ',//,12(' ',I2,' = ',E11.4,/)
     $        ,//)
  427 FORMAT (12X,2E11.4)
C**********
C
C
429   CONTINUE
      IF (SNOW .LT. 1)  GO TO 450
C     READ WIND/RADIATION ONLY FOR SNOWMELT CALCULATIONS
        DO  430 DA = 1,31
  430   READ(NMET,4050) (IWIND(MN,DA), MN=1,12)
C
        DO 440 DA =1,31
  440   READ(NMET,4050) (IRAD(MN,DA), MN=1,12)
C
  450   IF (UNIT .EQ. -1) GO TO 500
C*****      INPUT UNITS METRIC--CONVERT TO ENGLISH     *****
        DO 480 DA=1,31
          DO 470 MN=1,12
            IEVAP(MN,DA) = IEVAP(MN,DA)*3.937
C     CONVERT MM TO IN X .01
            IF (SNOW.EQ.1) IWIND(MN,DA) = IWIND(MN,DA)*0.6214
C     CONVERT KM TO MI
              DO 460 IT=1,2
      TMPCON=1.8*ITEMP(MN,DA,IT)+32.0
460   ITEMP(MN,DA,IT)=IFIX(TMPCON+SIGN(0.5,TMPCON))
C     CONVERT DEG C TO DEG F
 470          CONTINUE
 480        CONTINUE
C
C
C
 500    DO 1240  MONTH=MNSTRT,MNEND                 BEGIN MONTHLY LOOP
C
C           ASSIGN CURRENT MONTHLY VALUES OF ACCUMULATION RATES,
C           REMOVAL RATES, AND POTENCY FACTORS
C
      IF (HYCAL.EQ.1) GO TO 530
      IF (MNVAR.EQ.0.AND.MONTH.NE.BGNMON) GO TO 530
      DO 520 I=1,NLAND
      DO 510 J=1,NQUAL
      PMP(J,I)=PMPTAB(J,I,MONTH)
      PMD(J,I)=PMDTAB(J,I,MONTH)
  510 PMI(J,I)=PMITAB(J,I,MONTH)
      ACCP(I)=ACUPM(I,MONTH)
      ACCI(I)=ACUIM(I,MONTH)
      RPER(I)=REPERM(I,MONTH)
  520 RIMP(I)=REIMPM(I,MONTH)
  530 CONTINUE
      TEMPCF=TCF(MONTH)
C
C
C                   ZEROING OF VARIABLES
C
      DO 540 I=1,28
C        ZEROING OF THE FIRST 28 VARIABLES CONTAINED IN COMMON/LNDOUT/
  540 AR1OUT(I)=0.0
```

```
      PRTM=0.                                                           559.
      ROBTOM=0.                                                         560.
      INFTOM=0.                                                         561.
      DO 560 J=1,NQUAL                                                  562.
      DO 550 I=1,NLAND                                                  563.
      APOLP(I,J)=0.0                                                    564.
      APOLI(I,J)=0.0                                                    565.
  550 CONTINUE                                                          566.
      POLTCA(J)=0.0                                                     567.
  560 POLTC(J)=0.0                                                      568.
      DO 570 I=1,NLAND                                                  569.
      AERSN(I)=0.0                                                      570.
      AEIM(I)=0.0                                                       571.
  570 CONTINUE                                                          572.
      NOSIM=0                                                           573.
      NOS=0                                                             574.
      TEMPA=0.0                                                         575.
      DOA=0.0                                                           576.
      SEDTCA=0.0                                                        577.
      IX=2*MONTH                                                        578.
      IZ=IX-1                                                           579.
            DYSTRT = 1                                                  581.
      DYEND = DPM(MONTH)
      IF(MONTH.EQ.2.AND.LEAPYR) DYEND = DYEND+1
      IF(YEAR.EQ.BGNYR.AND.MONTH.EQ.BGNMON) DYSTRT=BGNDAY
      IF(YEAR.EQ.ENDYR.AND.MONTH.EQ.ENDMON) DYEND=ENDDAY
C                                                 BEGIN DAILY LOOP      603.
      IF(IDBUG) WRITE(ILP,6010) YEAR,MONTH,DYSTRT,DYEND
6010  FORMAT(' *****NPS MAIN--STARTING DAY LOOP IN ',I5,1X,I5,
     $' START=',I5,' END=',I5)
  660       DO 990  DAY=DYSTRT,DYEND                                    604.
      RECOUT(1)=MOD(YEAR,100)
      RECOUT(2)=MONTH
      RECOUT(3)=DAY
C     SAVE DATE FOR FILE OUTPUT
C                                                                       604.1
               IF (MONTH.EQ.1 .AND. DAY.EQ.1) JCOUNT = 0                604.2
               JCOUNT = JCOUNT + 1                                      604.3
C                                                                       604.4
               TIME = 0                                                 605.
               RAINT = 0.0                                              606.
               EP = PETMUL*IEVAP(MONTH,DAY)/1000.                       607.
               DO 670  I=1,INTRVL                                       608.
                  IRAIN(I) = 0                                          609.
                  RAIN(I) = 0.0                                         610.
  670             CONTINUE                                              611.
C                                                                       611.01
      MTX=MONTH                                                         611.02
      NTX=MONTH+1                                                       611.03
      IF (NTX.GT.12) NTX=1                                              611.04
C                                                                       611.05
C       CALCULATE EPXMI FOR EACH DAY AS LINEAR INTERPOLATION            611.06
C       BETWEEN MONTHLY VALUES                                          611.07
C                                                                       611.08
      EPXMI=EPXM(MTX)+(EPXM(NTX)-EPXM(MTX))*(FLOAT(DAY-1)/              611.09
     *      FLOAT(DYEND))                                               611.1
C          CHECK FOR TILLAGE DATES AND RESET SGEN                       612.2
```

```
      J=1
      DO 676 I=1,NLAND
      DO 674 II=1,12
      IF (JCOUNT.NE.TILDAY(I,II))  GO TO 674
      SGEN(I) =TILSED(I,II)
      IF (J.GT.0) WRITE(ILP,4610)
 4610 FORMAT(1H1)
      J=0
      ADEP=SGEN(I)
      IF (UNIT.GT.0) ADEP=ADEP*2.24
      WRITE(ILP,4600) MNAM(IZ),DAY,(LNDUSE(IK,I),IK=1,3),
     *                ADEP,WHT,ARUN
  674 CONTINUE
  676 CONTINUE
      IF(.NOT.DOT) GO TO 685
C*****      READ IN DAILY ACCUMULATION INPUTS      *****
      DO 680 I=1,NLAND
      ACCP(I)=IACCP(I,MONTH,DAY)
      ACCI(I)=IACCI(I,MONTH,DAY)
C           NO CONVERSION -- DIDOT INPUTS IN TONS/ACRE
      IF(UNIT.EQ.-1) GO TO 680
      ACCP(I)=IACCP(I,MONTH,DAY)*KGPHA/2000.0
      ACCI(I)=IACCI(I,MONTH,DAY)*KGPHA/2000.0
C     CONVERT FROM KG/HECTARE TO TONS/ACRE
  680 CONTINUE
  685 CONTINUE
C
C           CHECK TO SEE IF SNOWMELT CALC'S WILL BE DONE - IF YES THEN
C           CALCULATE CONTINUOUS TEMP, WIND, RAD AND APPLY CORRES MULT
C           FACTORS
C
      IF (SNOW.LT.1) GO TO 790
      WINF=(1.0-F) + F*(.35-.03*KUGI)
C
C                                   WINF REDUCES WIND FOR FORESTED AREAS
C
C          /* KUGI IS INDEX TO UNDERGROWTH AND FOREST DENSITY,*/
C          /* WITH VALUES 0 TO 10 - WIND IN FOREST IS 35% OF  */
C          /* WIND IN OPEN WHEN KUGI=0, AND 5% WHEN KUGI=10 - */
C          /* WIND IS ASSUMED MEASURED AT 1-5 FT ABOVE GROUND */
C          /* OR SNOW SURFACE  */
C
             WIND = IWIND(MONTH,DAY)
      TMIN = ITEMP(MONTH,DAY,2)
      DEWX = TMIN - 1.0*ELDIF
      RR = IRAD(MONTH,DAY)
C
C                             DEWPT ASSUMED TO BE MIN TEMP AND USES
C                             LAPSE RATE OF 1 DEGREE/1000 FT
C
      CALCULATE CONTINUOUS WIND AND RAD
      DO 780 I=1,24
        WINDX(I)  = WMUL*WIND*WINF*WINDIS(I)
        RAD(I)    = RMUL*RR*RADCON*RADDIS(I)
  780 CONTINUE
C
  790         IF (PINT.EQ.1)  GO TO 850         15-MIN PRECIP INPUT
      DO 830 J=1,8
              JK = J*12
```

```
                  JJ = JK - 11
                  READ(NMET,4020)  YR, MO, DY, CN, (IRAIN(I), I=JJ,JK)
                  IF (UNIT .EQ. -1) GO TO 820
C     CHANGE MM TO IN*0.01
                  DO 810 I=JJ,JK
810                  IRAIN(I) = IRAIN(I)*3.937 + 0.5
820               YR = YR + 1900
      IF(YEAR.EQ.YR.AND.MONTH.EQ.MO.AND.DAY.EQ.DY.AND.
     $(J.EQ.CN.OR.CN.EQ.9)) GO TO 825
                  WRITE(ILP,4000)  J, MONTH, DAY, YEAR, CN, MO, DY, YR
                  GO TO 1600
825   IF(CN.EQ.9) GO TO 835
830   CONTINUE
835            DO 840  I=1,INTRVL
                  RAIN(I) = IRAIN(I)*K1/100.
                  RAINT = RAINT + RAIN(I)
  840             CONTINUE
               GO TO 920
C
C                                           HOURLY PRECIP INPUT
850   DO 890 J=1,2
                  JK = J*48
                  JJ = JK - 47
                  READ(NMET,4020)  YR, MO, DY, CN, (IRAIN(I), I=JJ,JK,4)
                  IF (UNIT .EQ. -1) GO TO 880
C     CHANGE MM TO IN*0.01
                  DO 870 I=JJ,JK,4
870                  IRAIN(I) =IRAIN(I)*3.937 + 0.5
880               YR = YR + 1900
      IF(YEAR.EQ.YR.AND.MONTH.EQ.MO.AND.DAY.EQ.DY.AND.
     $(J.EQ.CN.OR.CN.EQ.9)) GO TO 885
                  WRITE(ILP,4000)  J, MONTH, DAY, YEAR, CN, MO, DY, YR
                  GO TO 1600
885   IF(CN.EQ.9) GO TO 895
890   CONTINUE
C
895            DO 910  I=1,INTRVL,4
                  TEM = IRAIN(I)*(K1/100.)/4.
                  DO 900 K=1,4
  900             RAIN(I+4-K)=TEM
                  RAINT = RAINT + RAIN(I)
  910             CONTINUE
C
920   CONTINUE
C*****     COMPUTE 24 HOUR TEMP DISTRIBUTION     *****
      MINTMP=ITEMP(MONTH,DAY,2)
      MAXTMP=ITEMP(MONTH,DAY,1)
      IF(DAY.NE.DYEND) MINNXT=ITEMP(MONTH,DAY+1,2)
      IF(DAY.EQ.DYEND.AND.MONTH.LT.12)
     $MINNXT=ITEMP(MONTH+1,1,2)
      IF(DAY.EQ.DYEND.AND.MONTH.EQ.12)
     $MINNXT=MINTMP
C     ASSUME NEXT DAY'S MIN = THIS DAY'S MIN AT END OF YEAR
      CALL TDIST(TEMPX,GRAD,MAXPRV,MINTMP,MAXTMP,MINNXT)
      MAXPRV=MAXTMP
C     UPDATE PREVIOUS DAY'S MAX TEMP
C**********
```

```
      IF (.NOT.EXAMS) GO TO 927
      FLOWEX=0.
      DO 925 M=1,2
      WETTMP(M)=0.
  925 POLEX(M)=0.
C
  927 CONTINUE
C**********
      IF(RAINT.GT.0.0) GO TO 950
C
C  USE RAIN LOOP IF MOISTURE STORAGES ARE NOT EMPTY
C
  930  IF ((RESS .LT. 0.001).OR.(SRGXT .LT. 0.001)) GO TO 980
C
C                   RAIN LOOP
C
C       CONDITIONAL BRANCHING TO CALCULATE HOURLY TEMPERATURES
C
  950  CONTINUE
C**********
      IF ((.NOT.DOT).OR.(RAINT.EQ.0.)) GO TO 959
C             HERE TO READ WET DEPOSITION
      NTOT=1+NQUAL
      DO 953 JQUAL=1,NTOT
  953 READ (ILUDOT,958) (WETDEP(IHR,JQUAL),IHR=1,24)
      DO 954 JHR=1,24
      DO 954 JQUAL=1,2
C             37.8=KG-DAY/TON-HOUR
  954 WETTMP(JQUAL)=WETTMP(JQUAL)+WETDEP(JHR,JQUAL)*AREA*37.8
      IF (IDBUG) WRITE (ILP,956) WETDEP
      IF (IDBUG) WRITE (ILP,957) WETTMP
  956 FORMAT (/' DIDOT WET DEPOSITIONS',/,
     $         ' SEDIMENT: ',12E10.3,/,11X,12E10.3,/,
     $         ' TOXICANT: ',12E10.3,/,11X,12E10.3)
  957 FORMAT (' DAILY TOTALS:  SED=',E11.4,' POL=',E11.4)
  958 FORMAT (12X,24E11.4)
C
  959 CONTINUE
C**********
C
C       CALCULATE COVER FUNCTION  FOR THE PERVIOUS
C             AREAS WITHiN EACH LAND TYPE USE
C
      MTX=MONTH
      NTX=MONTH+1
      IF (NTX.GT.12) NTX=1
      DO 960 I=1,NLAND
      COVER(I)=COVMAT(I,MTX)+(FLOAT(DAY-1)/FLOAT(DYEND))*
     1         (COVMAT(I,NTX)-COVMAT(I,MTX))
  960 CONTINUE
               DO 970  I=1,INTRVL
C**********
      DO 962 ILAND=1,NLAND
      TOTROF(ILAND)=0.
      SEDCON(ILAND)=0.
  962 POLCON(ILAND,1)=0.
C**********
```

```
            TIME = TIME + 1
            TF = 1
            PR = RAIN(I)
C
            IMIN = MOD(TIME,H)
            IHR = (TIME - IMIN)/H
            IMIN = TIMFAC*IMIN
            IX = 2*MONTH
            IZ = IX - 1
           CALL LANDS
    STEPS=INTRVL
    IF(HYCAL.NE.1) CALL QUAL
C**********
    IF (.NOT.EXAMS) GO TO 970
    DO 965 M=1,NLAND
C
C  CHANGE RUNOFF TO CUBIC METERS IF NECESSARY
    IF (UNIT.EQ.-1) TOTROF(M)=35.31*TOTROF(M)
C
C  WEIGHT CONCENTRATION OF SEDIMENT, POLEX(1), AND POLLUTANT POLEX(2)
C  BY RUNOFF TO GET CUBIC METERS-GRAMS/LITER-SECOND OF POLLUTANT WASHOFF
C  AND FACTOR BY 900*.000001 TO CONVERT TO KILOGRAMS OF POLLUTANT.
C  ALSO ADD ON WET DEPOSITION FOR THE DAY, IF ANY
    FLOWEX=FLOWEX+TOTROF(M)*900.
    POLEX(1)=POLEX(1)+(1.E-6)*900.*SEDCON(M)*TOTROF(M)+WETTMP(1)
    POLEX(2)=POLEX(2)+(1.E-6)*900.*POLCON(M,1)*TOTROF(M)+WETTMP(2)
C
 965 CONTINUE
C**********
C
C
 970 CONTINUE
            NDSR=0
C
C**********
    IF (EXAMS) WRITE (NOUT,999) (RECOUT(M),M=1,3),POLEX,FLOWEX
C**********
C
          GO TO 990
C
C             NO RAIN LOOP
C
  980     TF = INTRVL
          PR = 0.0
          P3 = 0.0
            RESB1 = 0.0
          IMIN = 00
          IHR = 24
          IX = 2*MONTH
          IZ = IX - 1
           NDSR=NDSR+1
           CALL LANDS
    STEPS=1.0
C   NUMBER OF INTERVALS IN NON-RAIN DAY = 1
    IF(HYCAL.NE.1) CALL QUAL
C   BRANCH TO QUALITY SUBROUTINE EXCEPT FOR HYDROLOGIC CALIBRATION RUNS
C**********
```

```
      IF (EXAMS) WRITE (NOUT,999) (RECOUT(M),M=1,3),POLEX,FLOWEX
  999 FORMAT (3I2,3E11.4)
C**********
C                                          END DAILY LOOP
  990         CONTINUE
C
C                           MONTHLY SUMMARY
C
      WRITE(ILP,4320) MNAM(IZ),MNAM(IX),YEAR
      UZSOT=UZS
      LZSOT=LZS
      SGWOT=SGW
      SCEPOT=SCEP
      RESSOT=RESS
      SRGXTO=SRGXT
      TWBALO=TWBAL
      TSNBOL=TSNBAL
      PACKOT=PACK
      IF (UNIT.EQ.-1) GO TO 1010
      DO 1000 I=1,28
C          CONVERSION TO METRIC UNITS OF THE FIRST 28 VARIABLES
C           CONTAINED IN COMMON/LNDOUT/
      AR1OUT(I)=AR1OUT(I)*MMPIN
 1000 CONTINUE
 1010 WRITE(ILP,4330) DEPW,ROSTOM,RINTOM,RITOM,BASTOM,RUTOM,RCHTOM,PRTOM
      IF (SNOW.LT.1) GO TO 1030
      COVR=100.
      IF (PACK.LT.IPACK) COVR=(PACK/IPACK)*100.
      IF (PACK.GT.0.01) GO TO 1020
      COVR=0.0
      SDEN=0.0
 1020 WRITE(ILP,4340) SUMSNM,PXSNM,MELRAM,RADMEM,CONMEM,CDRMEM,CRAINM,
     *                SGMM,SNEGMM,PACKOT,SDEN,COVR,SEVAPM
 1030 WRITE(ILP,4350) EPTOM,NEPTOM,UZSOT,LZSOT,SGWOT,SCEPOT,RESSOT,
     *                SRGXTO,TWBALO
      IF (SNOW.GT.0) WRITE(ILP,4360) TSNBOL
        IF (HYCAL.EQ.1) GO TO 1230
C
C          OUTPUT OF SEDIMENTS DEPOSIT ON GROUND AT MONTH'S END
C
      WRITE(ILP,4370) WHT,ARUN
       TEM1=0.0
       TEM2=0.0
       TEM3=0.0
       TEM4=0.0
      TEM8=0.0
      DO 1050 I=1,NLAND
      STOT=SRER(I)+SGEN(I)
C     TOTAL PERVIOUS SEDIMENTS
      TEM=STOT*(1-IMPK(I))+TS(I)*IMPK(I)
      WHFUN1=(AR(I)/AREA)*(1-IMPK(I))
      WHFUN2=(AR(I)/AREA)*IMPK(I)
      TEM1=TEM1+SGEN(I)*WHFUN1
      TEM8=TEM8+SRER(I)*WHFUN1
      TEM2=TEM2+TS(I)*WHFUN2
      TEM3=TEM3+WHFUN1
      TEM4=TEM4+WHFUN2
      IF (UNIT.GT.-1) GO TO 1040
      WRITE(ILP,4390) (LNDUSE(IK,I),IK=1,3),TEM,SGEN(I),SRER(I),TS(I)
      GO TO 1050
 1040 TEM5=SRER(I)*2.24
      TEM7=SGEN(I)*2.24
      TEM6=TS(I)*2.24
      TEM=TEM*2.24
C     CONVERT TONS/ACRE TO METRIC TONS/HA
      WRITE(ILP,4390) (LNDUSE(IK,I),IK=1,3),TEM,TEM7,TEM5,TEM6
 1050 CONTINUE
      IF (NLAND.EQ.1) GO TO 1070
      IF (TEM3.GT.0.0) TEM1=TEM1/TEM3
      IF(TEM3.GT.0.0) TEM8=TEM8/TEM3
      IF (TEM4.GT.0.0) TEM2=TEM2/TEM4
      TEM=(TEM1+TEM8)*(1-A)+TEM2*A
      IF (UNIT.LT.1) GO TO 1060
      TEM=TEM*2.24
      TEM1=TEM1*2.24
      TEM2=TEM2*2.24
      TEM8=TEM8*2.24
 1060 WRITE(ILP,4380) TEM,TEM1,TEM8,TEM2
C
C           OUTPUT MONTHLY SEDIMENTS LOSS FOR EACH LAND TYPE USE
C
 1070 WRITE(ILP,4400) WHT,WHGT,ARUN
      AERSNT=0.0
      AEIMT=0.0
      DO 1100 I=1,NLAND
      TEM=AEIM(I)+AERSN(I)
C     AERSN AND AEIM ARE TOTALS FOR LAND USE AREA
      IF (TEM.GT.0.0) GO TO 1080
      TEM1=0.0
      TEM2=0.0
      TEM3=0.0
      GO TO 1090
 1080 TEM1=TEM*2000./AR(I)
C     COMPUTE LBS/ACRE
      TEM2=100.*AERSN(I)/TEM
      TEM3=100.*AEIM(I)/TEM
      IF (UNIT.LT.1) GO TO 1090
      TEM=TEM*.9072
C     CONVERT TONS TO METRIC TONS
      TEM1=TEM1/KGPHA
 1090 WRITE(ILP,4410) (LNDUSE(IK,I),IK=1,3),TEM,TEM1,TEM2,TEM3
      AERSNT=AERSNT+AERSN(I)
      AEIMT=AEIMT+AEIM(I)
 1100 CONTINUE
C
C           OUTPUT MONTHLY SEDIMENTS LOSS FOR THE ENTIRE WATERSHED
C
      TEM=AERSNT+AEIMT
      IF (TEM.GT.0.0) GO TO 1110
      TEM1=0.0
      TEM2=0.0
      TEM3=0.0
      GO TO 1120
 1110 TEM1=TEM*2000./AREA
```

```
      TEM2=100.*AERSNT/TEM
      TEM3=100.*AEIMT/TEM
      IF (UNIT.LT.1) GO TO 1120
      TEM=TEM*.9072
      TEM1=TEM1/KGPHA
 1120 WRITE(ILP,4470) TEM,TEM1,TEM2,TEM3
      WRITE(ILP,4420) WHGT ,WHGT,ARUN
C
C           OUTPUT MONTHLY WASHOFF FOR EACH OF THE ANALYZED POLLUTANTS
C
      DO 1180 J=1,NQUAL
      WRITE(ILP,4430) (QUALIN(I,J),I=1,3)
      APOLPT=0.0
      APOLIT=0.0
      DO 1150 I=1,NLAND
C
C           MONTHLY WASHOFF OF A GIVEN POLLUTANT FROM EACH LAND TYPE USE
C
      TEM=APOLP(I,J)+APOLI(I,J)
      IF (TEM.GT.0.0) GO TO 1130
      TEM1=0.0
      TEM2=0.0
      TEM3=0.0
      GO TO 1140
 1130 TEM1=TEM/AR(I)
      TEM2=100.*APOLP(I,J)/TEM
      TEM3=100.*APOLI(I,J)/TEM
      IF (UNIT.LT.1) GO TO 1140
      TEM=TEM*.454
      TEM1=TEM1/KGPHA
 1140 WRITE(ILP,4410) (LNDUSE(KK,I),KK=1,3),TEM,TEM1,TEM2,TEM3
      APOLPT=APOLPT+APOLP(I,J)
      APOLIT=APOLIT+APOLI(I,J)
 1150 CONTINUE
C
C          TOTAL MONTHLY WASHOFF OF A GIVEN POLLUTANT
C
      TEM=APOLPT+APOLIT
      IF (TEM.GT.0.0) GO TO 1160
      TEM1=0.0
      TEM2=0.0
      TEM3=0.0
      GO TO 1170
 1160 TEM1=TEM/AREA
      TEM2=100.*APOLPT/TEM
      TEM3=100.*APOLIT/TEM
      IF (UNIT.LT.1) GO TO 1170
      TEM=TEM*.454
      TEM1=TEM1/KGPHA
 1170 WRITE(ILP,4440) TEM,TEM1,TEM2,TEM3
 1180 CONTINUE
      TEMPAY=TEMPAY+TEMPA
      DOAY=DOAY+DOA
      SEDTCY=SEDTCY+SEDTCA
C
C           CALCULATE AND PRINT MONTHLY AVERAGES OF TEMPERATURE,
C           DISSOLVED OXYGEN,AND EACH OF THE ANALYSED POLLUTANT
```

```
C
      IF (NOSIM.LE.0) GO TO 1190
      TEMPA=TEMPA/NOSIM
      DOA=DOA/NOSIM
      SEDTCA=SEDTCA/NOSIM
 1190 TEMPO=TEMPA
      IF (UNIT.EQ.1) TEMPO=(TEMPO-32.)*5/9
      WRITE(ILP,4450) UTMP,TEMPO,DOA,SEDTCA
      DO 1210 J=1,NQUAL
      PLTCAY(J)=PLTCAY(J)+POLTCA(J)
      IF (NOSIM.LE.0) GO TO 1200
      POLTCA(J)=POLTCA(J)/NOSIM
 1200 WRITE(ILP,4460) (QUALIN(I,J),I=1,3),CUNIT(J),POLTCA(J)
 1210 CONTINUE
C
C              ACCUMULATION FOR YEARLY SUMMARIES
C
      DO 1220 I=1,NLAND
      AERSNY(I)=AERSNY(I)+AERSN(I)
      AEIMY(I)=AEIMY(I)+AEIM(I)
      DO 1220 J=1,NQUAL
      APOLPY(I,J)=APOLPY(I,J)+APOLP(I,J)
      APOLIY(I,J)=APOLIY(I,J)+APOLI(I,J)
 1220 CONTINUE
 1230 CONTINUE
      WRITE(ILP,4490) NOS
      NOSIY=NOSIY+NOSIM
      NOSY=NOSY+NOS
 1240 CONTINUE
C                                   END MONTHLY LOOP
C                              YEARLY SUMMARIES
C
      WRITE(ILP,4480) YEAR
      UZSMT=UZS
      LZSMT=LZS
      SGWMT=SGW
      SCEPT=SCEP
      RESST=RESS
      SRGXTT=SRGX
      TWBLMT=TWBAL
      TSNBOL=TSNBAL
      IF (UNIT.EQ.-1) GO TO 1260
      DO 1250 I=1,28
C         CONVERSION TO METRIC UNITS OF THE LAST 28 VARIABLES
C          CONTAINED IN COMMON/LNDOUT/
 1250 AR2OUT(I)=AR2OUT(I)*MMPIN
 1260 WRITE(ILP,4330) DEPW,ROSTOT,RINTOT,RITOT,BASTOT,RUTOT,RCHTOT,PRTOT
      IF (SNOW.LT.1) GO TO 1280
      COVR=100.
      IF (PACK.LT.IPACK) COVR=(PACK/IPACK)*100.
      IF (PACK.GT.0.01) GO TO 1270
      COVR=0.0
      SDEN=0.0
 1270 WRITE(ILP,4340) SUMSNY,PXSNY,MELRAY,RADMEY,CONMEY,CDRMEY,CRAINY,
     *               SGMY,SNEGMY,PACKOT,SDEN,COVR,SEVAPY
 1280 WRITE(ILP,4350) EPTOT,NEPTOT,UZSMT,LZSMT,SGWMT,SCEPT,RESST,
     *               SRGXTT,TWBLMT
```

```
      IF (SNOW.GT.0) WRITE(ILP,4360) TSNBOL
      IF (HYCAL.EQ.1) GO TO 1425
      WRITE(ILP,4400) WHT,WHT,ARUN
C
C        OUTPUT YEARLY SEDIMENTS LOSS FOR EACH LAND TYPE USE
C
      AERSNT=0.0
      AEIMT=0.0
      DO 1310 I=1,NLAND
      TEM=AEIMY(I)+AERSNY(I)
      IF (TEM.GT.0.0) GO TO 1290
      TEM1=0.0
      TEM2=0.0
      TEM3=0.0
      GO TO 1300
 1290 TEM1=TEM/AR(I)
      TEM3=100.*AEIMY(I)/TEM
      TEM2=100.*AERSNY(I)/TEM
      IF (UNIT.LT.1) GO TO 1300
      TEM=TEM*.9072
      TEM1=TEM1/KGPHA
 1300 WRITE(ILP,4410) (LNDUSE(KK,I),KK=1,3),TEM,TEM1,TEM2,TEM3
      AERSNT=AERSNT+AERSNY(I)
      AEIMT=AEIMT+AEIMY(I)
 1310 CONTINUE
C
C        OUTPUT YEARLY SEDIMENTS LOSS FOR THE ENTIRE WATERSHED
C
      TEM=AERSNT+AEIMT
      IF (TEM.GT.0.0) GO TO 1320
      TEM1=0.0
      TEM2=0.0
      TEM3=0.0
      GO TO 1330
 1320 TEM1=TEM/AREA
      TEM2=100.*AERSNT/TEM
      TEM3=100.*AEIMT/TEM
      IF (UNIT.LT.1) GO TO 1330
      TEM=TEM*.9072
      TEM1=TEM1*2.24
 1330 WRITE(ILP,4470) TEM,TEM1,TEM2,TEM3
      WRITE(ILP,4420) WHGT,WHGT,ARUN
C
C        OUTPUT YEARLY WASHOFF FOR EACH OF THE ANALYZED POLLUTANTS
C
      DO 1390 J=1,NQUAL
      WRITE(ILP,4430) (QUALIN(I,J),I=1,3)
      APOLPT=0.0
      APOLIT=0.0
      DO 1360 I=1,NLAND
C
C        YEARLY WASHOFF OF A GIVEN POLLUTANT FROM EACH LAND TYPE USE
C
      TEM=APOLPY(I,J)+APOLIY(I,J)
      IF (TEM.GT.0.0) GO TO 1340
      TEM1=0.0
      TEM2=0.0
      TEM3=0.0
      GO TO 1350
 1340 TEM1=TEM/AR(I)
      TEM2=100.*APOLPY(I,J)/TEM
      TEM3=100.*APOLIY(I,J)/TEM
      IF (UNIT.LT.1) GO TO 1350
      TEM=TEM*.454
      TEM1=TEM1/KGPHA
 1350 WRITE(ILP,4410) (LNDUSE(KK,I),KK=1,3),TEM,TEM1,TEM2,TEM3
      APOLPT=APOLPT+APOLPY(I,J)
      APOLIT=APOLIT+APOLIY(I,J)
 1360 CONTINUE
C
C        TOTAL YEARLY WASHOFF OF A GIVEN POLLUTANT
C
      TEM=APOLPT+APOLIT
      IF (TEM.GT.0.0) GO TO 1370
      TEM1=0.0
      TEM2=0.0
      TEM3=0.0
      GO TO 1380
 1370 TEM1=TEM/AREA
      TEM2=100.*APOLPT/TEM
      TEM3=100.*APOLIT/TEM
      IF (UNIT.LT.1) GO TO 1380
      TEM=TEM*.454
      TEM1=TEM1/KGPHA
 1380 WRITE(ILP,4440) TEM,TEM1,TEM2,TEM3
 1390 CONTINUE
C
C         CALCULATE AND PRINT YEARLY AVERAGES OF TEMPERATURE,
C         DISSOLVED OXYGEN,AND EACH OF THE ANALYZED POLLUTANT
C
      IF (NOSIY.LE.0) GO TO 1400
      TEMPAY=TEMPAY/NOSIY
      DOAY=DOAY/NOSIY
      SEDTCY=SEDTCY/NOSIY
 1400 TEMPO=TEMPAY
      IF (UNIT.EQ.1) TEMPO=(TEMPO-32.)*5/9
      WRITE(ILP,4450) UTMP,TEMPO,DOAY,SEDTCY
      DO 1420 J=1,NQUAL
      IF (NOSIY.LE.0) GO TO 1410
      PLTCAY(J)= PLTCAY(J)/NOSIY
 1410 WRITE(ILP,4460) (QUALIN(I,J),I=1,3),CUNIT(J),PLTCAY(J)
 1420 CONTINUE
 1425 WRITE(ILP,4490) NOSY
C
C                       ZEROING OF VARIABLES
C
      DO 1430 I=1,28
C         ZEROING OF THE LAST 28 VARIABLES CONTAINED IN COMMON/LNDOUT/
 1430 AR2OUT(I)=0.0
      DO 1450 J=1,NQUAL
      DO 1440 I=1,5
      APOLPY(I,J)=0.0
 1440 APOLIY(I,J)=0.0
 1450 PLTCAY(J)=0.0
```

```
      DO 1460 I=1,5
      AERSNY(I)=0.0
 1460 AEIMY(I)=0.0
      NOSIY=0
      TEMPAY=0.0
      DOAY=0.0
C
C         SUMMARY OF STORMS' CHARACTERISTICS
C
      NV=(NQUAL+1)*4
      IF (HYCAL.EQ.1) NV=2
      IF (NOSY.LT.2.OR.NOSY.GT.200) GO TO 1560
C
C         CLEAR OUTPUT VECTORS AND INITIALIZE VMIN AND VMAX
C
      DO 1470 K=1,NV
      TOTAL(K)=0.0
      SD(K)=0.0
      VMIN(K)=STMCH(1,K)
 1470 VMAX(K)=STMCH(1,K)
C
C         CALCULATE MEANS, ST.DEV'S,MAXIMA, AND MINIMA
C
      DO 1520 I=1,NOSY
      DO 1520 K=1,NV
      TOTAL (K)=TOTAL(K)+STMCH(I,K)
      IF (STMCH(I,K)-VMIN(K)) 1480,1490,1490
 1480 VMIN(K)=STMCH(I,K)
 1490 IF (STMCH(I,K)-VMAX(K)) 1510,1510,1500
 1500 VMAX(K)=STMCH(I,K)
 1510 SD(K)=SD(K)+STMCH(I,K)*STMCH(I,K)
 1520 CONTINUE
      DO 1530 K=1,NV
      RANGE(K)=VMAX(K)-VMIN(K)
      AVER(K)=TOTAL(K)/NOSY
      SD(K)=SQRT((SD(K)-TOTAL(K)*TOTAL(K)/NOSY)/(NOSY-1))
 1530 CONTINUE
C
C         PRINT STORM CHARACTERISTICS
C
      IF (HYCAL.NE.1) GO TO 1540
      WRITE(ILP,4500)
      WRITE(ILP,4580) DEPW,AVER(1),SD(1),VMAX(1),VMIN(1),RANGE(1)
      WRITE(ILP,4590) UFL,AVER(2),SD(2),VMAX(2),VMIN(2),RANGE(2)
      GO TO 1570
 1540 WRITE(ILP,4500)
      WRITE(ILP,4510)
      WRITE(ILP,4530) WHT,AVER(1),SD(1),VMAX(1),VMIN(1),RANGE(1)
      WRITE(ILP,4540) WHGT,AVER(2),SD(2),VMAX(2),VMIN(2),RANGE(2)
      WRITE(ILP,4545) AVER(3),SD(3),VMAX(3),VMIN(3),RANGE(3)
      WRITE(ILP,4555) AVER(4),SD(4),VMAX(4),VMIN(4),RANGE(4)
      DO 1550 J=1,NQUAL
      WRITE(ILP,4430) (QUALIN(I,J),I=1,3)
      K=J*4+1
      WRITE(ILP,4530) WHT,AVER(K),SD(K),VMAX(K),VMIN(K),RANGE(K)
      K=K+1
      WRITE(ILP,4540) WHGT,AVER(K),SD(K),VMAX(K),VMIN(K),RANGE(K)
      K=K+1
      WRITE(ILP,4550) CUNIT(J),AVER(K),SD(K),VMAX(K),VMIN(K),RANGE(K)
      K=K+1
      WRITE(ILP,4560) CUNIT(J),AVER(K),SD(K),VMAX(K),VMIN(K),RANGE(K)
 1550 CONTINUE
      GO TO 1570
 1560 WRITE(ILP,4570)
      IF (NOSY.EQ.0) GO TO 1590
 1570 DO 1580 I=1,NOSY
      DO 1580 K=1,NV
 1580 STMCH(I,K)=0.0
      NOSY=0
C                                END OF YEARLY LOOP
 1590    CONTINUE
C
 1600 CONTINUE
      STOP
C
C                     FORMAT STATEMENTS
C
 4000 FORMAT ('1','*****ERROR*****  INCORRECT INPUT DATA!   DESIRED ',
     *   'CARD ',I1,' FOR ',I2,'/',I2,'/',I4,'; READ CARD ',I1,' FOR ',
     *   I2,'/',I2,'/',I4)
 4010 FORMAT ('0')
 4020 FORMAT (1X,3I2,I1,12I6)
 4040 FORMAT (8X,24I3)
 4050 FORMAT (8X,12I6)
 4060 FORMAT (3A4,2X,A4 )
 4070 FORMAT ('1',9(/),45X,'NONPOINT SOURCE POLLUTANT LOADING MODEL',
     1        /,44X,42('='),5(/))
 4080 FORMAT (' ',1X,'WATERSHED CHARACTERISTICS :',///,6X,'NAME',8X,
     1        3A8,/,18X,3A8,//,6X,'TOTAL AREA (',A4,')',8X,F9.2,/)
 4090 FORMAT (9X,'LAND USE',5X,'% OF TOTAL',6X,'AREA (',A4,'S)',6X,
     1        'PERVIOUS (',A4,'S)',3X,'IMPERVIOUS (',A4,'S)',3X,
     2        'IMPERVIOUS (%)',/)
 4100 FORMAT (' ',7X,3A4,5X,F5.1,4(10X,F9.2))
 4110 FORMAT (/,6X,'FRACTION OF IMPERVIOUS AREA',2X,F5.2)
 4120 FORMAT ('0',8X,'**WARNING**',3X,'CHECK IF THE LAND TYPES AREAS ',
     1        'ARE CORRECT')
 4130 FORMAT (5(/),' ',1X,'SIMULATION CHARACTERISTICS :',///,6X,
     1        'TYPE OF RUN',10X,4A8,/,6X,'DATE SIMULATION BEGINS',
     2        13X,2A4,2X,I2,', ',I4,/,6X,'DATE SIMULATION ENDS',15X,
     3        2A4,2X,I2,', ',I4,/,6X,'INPUT PRECIPITATION TIME INTERVAL',
     4        9X,I3,1X,'MINUTES',/,6X,'SIMULATION TIME INTERVAL',19X,I2,
     5        1X,'MINUTES',/,6X,'IS SNOWMELT CONSIDERED ?',26X,A4,/,
     6        6X,'INPUT UNITS',34X,1A8,/,6X,'OUTPUT UNITS',33X,1A8,/,6X,
     7        'MINIMUM FLOW FOR OUTPUT PER INTERVAL (',A4,')',1X,F9.4,
     8        /,6X,'NUMBER OF QUALITY INDICATORS ANALYZED',14X,I2,
     9        /,6X,'THE ANALYZED QUALITY INDICATORS',4X,
     X        'SEDIMENTS,DO,TEMP,',/,5(46X,3A4,',',/))
 4140 FORMAT (5(/),2X,'SUMMARY OF INPUT PARAMETERS :',///,6X,
     1         'LANDS',13X,'INTER =',F7.3,4X,'IRC    =',F8.3,3X,
     2         'INFIL =',F7.3,/,24X,'NN     =',F7.3,4X,'L      =',
     3         F8.3,3X,'SS     =',F7.3,/,24X,'NNI    =',F7.3,4X,
     $         'LI     =',F8.3,3X,'SSI    =',F7.3,/,24X,'K1     =',F7.3,
     5         4X,'PETMUL=',F8.3,3X,'K3     =',F7.3,/,24X,14X,
     6         4X,'K24L  =',F8.3,3X,'KK24  =',F7.3,/,24X,
      8         'UZSN  =',F8.3,3X,'LZSN  =',F8.3)
 4150 FORMAT (/,6X,'SNOW',14X,'RADCON=',F7.3,4X,'CCFAC =',F7.3,4X,
      1         'EVAPSN=',F8.3,/24X,'MELEV =',F8.3,3X,'ELDIF =',F8.3,3X,
      2         'TSNOW =',F7.3,/,24X,'MPACK =',F7.3,4X,'DGM   =',F7.3,
      3         4X,'WC    =',F7.3,/,24X,'IDNS  =',F7.3,4X,'SCF   =',F7.3,
      4         4X,'WMUL  =',F7.3,/,24X,'RMUL  =',F7.3,4X,'F     =',
      5         F7.3,4X,'KUGI  =',F7.3,/)
 4160 FORMAT (/,6X,'QUAL  ',12X,'JRER  =',F7.3,4X,'KRER  =',F7.3,/,
      1         24X,'JSER  =',F7.3,4X,'KSER  =',F7.3,/,
      2         24X,'JEIM  =',F7.3,4X,'KEIM  =',F7.3,/)
 4162 FORMAT (/,24X,'TIMTIL AND SRERTL-')
 4165 FORMAT (/,24X,3A4,12(5X,I3),/,36X,12(3X,F5.2))
 4170 FORMAT (//,6X,'MONTHLY DISTRIBUTION',7X,11(A4,4X),A4,//,6X,
      1      'TEMP CORRECTION FACTOR',1X,12(2X,F6.2),
      2     /,6X,'EPXM',19X,12(2X,F6.2),///,7X,
      3      '- PERVIOUS LANDS -',///,6X,'LAND COVER-',3A4,1X,12(1X,F7.3))
 4180 FORMAT (17X,3A4,1X,12(1X,F7.3))
 4190 FORMAT (/,6X,'ACCUMULATION RATES')
 4200 FORMAT (/,6X,'REMOVAL RATES')
 4210 FORMAT (/,6X,'POTENCY FACTORS FOR',1X,3A4)
 4220 FORMAT (//,6X,'-IMPERVIOUS LANDS-',/)
 4230 FORMAT (24X,3A4,6X,'ACUP  =',F7.3,4X,'ACUI  =',F7.3)
 4240 FORMAT (24X,3A4,6X,'RPER  =',F7.3,4X,'RIMP  =',F7.3)
 4250 FORMAT (//,6X,'POTENCY FACTORS FOR PERVIOUS AREAS(',A4,')',
     $5X,5(3A4,3X),/)
 4260 FORMAT (24X,3A4,8X,5(F8.3,7X))
 4270 FORMAT (//,6X,'POTENCY FACTORS FOR IMPERVIOUS AREAS',5(3X,3A4),/)
 4280 FORMAT (5(/),2X,'INITIAL CONDITIONS :',3(/),6X,'LANDS',13X,
      1         'UZS   =',F7.3,4X,'LZS   =',F7.3,4X,'SGW   =',F7.3,/)
 4290 FORMAT (6X,'SNOW',14X,'PACK  =',F7.3,4X,'DEPTH =',F7.3,/)
 4300 FORMAT (6X,'QUAL',14X,3A4,6X,'TSI =',F9.3,4X,'SRERI =',F9.3,
     $4X,'SGENI =',F9.3)
 4310 FORMAT (24X,3A4,6X,'TSI =',F9.3,4X,'SRERI =',F9.3,4X,'SGENI =',
     $F9.3)
 4320 FORMAT ('1',25X,'SUMMARY FOR MONTH OF ',2A4,1X,I4,/,
      1         25X,35('='),//,35X,'TOTAL')
 4330 FORMAT ('0',8X,'WATER   ',A4,//,11X,'RUNOFF',/,14X,
      1         'OVERLAND FLOW',5X,F9.3,/,14X,'INTERFLOW',9X,F9.3
      2         /,14X,'IMPERVIOUS',8X,F9.3,/,14X,'BASE FLOW',9X,
      3         F9.3,/,14X,'TOTAL',13X,F9.3,//,11X,
      4         'GRDWATER RECHARGE',4X,F9.3,//,11X,'PRECIPITATION',
      5         8X,F9.3)
 4340 FORMAT (' ',13X,'SNOW',14X,F9.3,/,14X,'RAIN ON SNOW',6X,
      1         F9.3,/,14X,'MELT & RAIN',7X,F9.3,//,11X,'MELT',
      2        /,14X,'RADIATION',9X,F9.3,/,14X,'CONVECTION',8X,
      3         F9.3,/,14X,'CONDENSATION',6X,F9.3,/,14X,'RAIN - MELT',
      4         7X,F9.3,/,14X,'GROUND-MELT',7X,F9.3,/,14X,
      5         'CUM-NEG-HEAT',6X,F9.3,//,11X,'SNOW-PACK',12X,F9.3,
      6         /,11X,'SNOW DENSITY',9X,F9.3,/,11X,'% SNOW COVER',
      7         9X,F9.3,//,11X,'SNOW EVAP',12X,F9.3)
 4350 FORMAT ('0',11X,'EVAPOTRANSPIRATION',/,14X,'POTENIAL',10X,
      1         F9.3,/,14X,'NET',15X,F9.3,//,11X,'STORAGES',/,
      2         14X,'UPPER ZONE',8X,F9.3,/,14X,'LOWER ZONE',8X,F9.3,
      3         /,14X,'GROUNDWATER',7X,F9.3,/,14X,'INTERCEPTION',6X,
      4         F9.3,/,14X,'OVERLAND FLOW',5X,F9.3,/,14X,'INTERFLOW',
      5         9X,F9.3,//,11X,'WATER BALANCE',8X,F9.3)
 4360 FORMAT (' ',10X,'SNOW BALANCE',9X,F9.3)
 4370 FORMAT ('0',8X,'SEDIMENTS ACCUMULATION ,',A4,'/',A4,9X,
     $'WEIGHTED MEAN',T72,'PERVIOUS',T92,'PERVIOUS',T112,
     $'IMPERVIOUS',/,T72,'GENERATED',T92,'DEPOSITED',/)
 4380 FORMAT (11X,'WEIGHTED MEAN',27X,E11.3,4(10X,E11.3))
 4390 FORMAT (' ',8X,3A4,29X,E11.3,4(9X,E11.3))
 4400 FORMAT ('0',8X,'SEDIMENTS LOSS, ',11X,'TOTAL (',A4,')',3X,
    1     'TOTAL (',A4,'/',A4,')',3X,'PERVIOUS (%)',7X,'IMPERVIOUS (%)')
 4410 FORMAT (' ',8X,3A4,9X,F11.3,3(5X,E11.3))
 4420 FORMAT ('0',8X,'POLLUTANT WASHOFF, ',8X,'TOTAL (',A4,')',3X,
    1     'TOTAL (',A4,'/',A4,')',3X,'PERVIOUS (%)',7X,'IMPERVIOUS (%)')
 4430 FORMAT ('0', 9X,'WASHOFF OF ',3A4)
 4440 FORMAT (' ',10X,'TOTAL WASHOFF',6X,F11.3,3(9X,F11.3))
 4450 FORMAT ('0',8X,'STORM WATER QUALITY - AVERAGES',//,
    1         11X,'TEMPERATURE ',A4,6X,F7.2,//,11X,
    2         'DISSOLVED OXYGEN (PPM)',1X,F7.3,//,12X,
    3         'SEDIMENTS   (GM/L)',F11.3)
 4460 FORMAT (' ',11X,3A4,'(',A4,')',F11.3)
 4470 FORMAT (' ',10X,'TOTAL LOSS',10X,F10.3,3(10X,F10.3))
 4480 FORMAT ('1',25X,'SUMMARY FOR ',I4,/,25X,16(' '),//,35X,'TOTAL')
 4490 FORMAT ('0',8X,'NO. OF STORMS',14X,I3)
 4500 FORMAT ('0',8X,'SUMMARY OF STORMS' CHARACTERISTICS',4X,
    1         'AVERAGE',8X,'ST.DEV.',9X,'MAXIMA',9X,'MINIMA',
    2       9X,'RANGE',//)
 4510 FORMAT (11X,'SEDIMENTS LOSS')
 4530 FORMAT (/,14X,'TOTAL WASHOFF (',A4,')',4X,5(5X,F10.3))
 4540 FORMAT (14X,'MAX WASHOFF (',A4,'/15MIN)',5X,F10.3,
    1         4(5X,F10.3))
 4545 FORMAT (14X,'MEAN CONCENTRATION (GM/L)',4X,F10.3,4(5X,F10.3))
 4550 FORMAT (14X,'MEAN CONCENTRATION (',A4,')',4X,F10.3,4(5X,F10.3))
 4555 FORMAT (14X,'MAX CONCENTRATION (GM/L)',5X,F10.3,4(5X,F10.3))
 4560 FORMAT (14X,'MAX CONCENTRATION (',A4,')',5X,F10.3,4(5X,F10.3 ))
 4570 FORMAT ('0',8X,'**WARNING**',3X,
    1         'SUMMARY OF STORM CHARACTERISTICS NOT PRINTED',
    2         /,22X,'NUMBER OF STORMS LESS THAN 2 OR MORE THAN 200',
    3         '   -  CHECK YOUR HYMIN PARAMETER')
 4580 FORMAT (/,14X,'TOTAL RUNOFF (',A4,')',5X,5(5X,F10.3))
 4590 FORMAT (14X,'MAX RUNOFF (',A4,')',12X,F10.3,
    1         4(5X,F10.3))
 4600 FORMAT (/,' *** ',A4,1X,I2,'  -  LAND SURFACE DISTURBANCE OCCURS
     * ON  ',3A4,/,17X,'GENERATED SEDIMENT ON PERVIOUS AREAS RESET TO',
     * F9.3,1X,A4,'/',A4)
C
      END
      BLOCK DATA                                                        2000.
C                 BLOCK DATA TO INITIALIZE VARIABLES                    2003.
      IMPLICIT  REAL(L)                                                 2007.
      LOGICAL LAST, PREV                                                2058.
C**********
      LOGICAL IDBUG,LDBUG,DOT,EXAMS
C**********
      INTEGER  UNIT , LMTS, RECOUT, SFLAG, PRINTR                       2056.
      INTEGER TIMTIL, TILDAY, DPM                                       2056.1
C                                                                       2059.
      REAL*8  WSNAME,RTYPE,UTYPE                                        2060.
      REAL  JRER, KRER, JSER, KSER,KEIM,JEIM                            2061.
      REAL  LZSN, IRC, NN, L, LZS, KV, K24L, KK24, INFIL, INTER         2062.
      REAL  IFS, K24EL, K3, NEPTOM, NEPTOT, ICS, NNI, KUGI              2063.
      REAL  INFTOM, INFTOT, INTF                                        2064.
      REAL  MMPIN, METOPT, KGPLB, KGPHA                                 2065.
      REAL IMPK
      REAL  MELRAM, MELRAY                                              2067.
C                                                                       2008.
      DIMENSION MNAM(24),RAD(24),TEMPX(24),WINDX(24),RAIN(96),          2009.
     1          GRAD(24),RADDIS(24),WINDIS(24)                          2010.
C                                                                       2011.
      COMMON /ALL/ RU,HYMIN,HYCAL,DPST,UNIT,TIMFAC,LZS,AREA,RESB,SFLAG, 2012.
     1             RESB1,ROSB,SRGX,INTF,RGX,RUZB,UZSB,PERCB,RIB,P3,TF,  2013.
     2             KGPLB,LAST,PREV,TEMPX,IHR,IHRR,PR,RUI,A,PA,GWF,NOSY, 2014.
     3             SRER(5),TS(5),LNDUSE(3,5),AR(5),QUALIN(3,5),NOSI,NOS,2015.
     4             NOSIM,UFL,UTMP,UNT1(2,2),UNT2(2,2),UNT3(2,2),WHGT,   2016.
     5             WHT,DEPW,ROSBI,RESBI,RESBI1,ARUN,LMTS(5),IMPK(5),    2017.
     6             NLAND,NQUAL,STMCH(200,24),RECOUT(5),FLOUT,SCALEF(5), 2018.
     7             SNOW,PACK,IPACK,SGEN(5)                              2019.
C                                                                       2020.
      COMMON /LAND/DAY,PRTM,IMIN,IX,TWBAL,SGW,GWS,KV,LIRC4,LKK4,ALTR(9),2021.
     1             UZS,IZ,UZSN,LZSN,INFIL,INTER,SGW1,DEC,DECI,TIT(13),  2022.
     2             K24L,KK24,K24EL,EP,IFS,K3,EPXMI,RESS1,RESS,SCEP,IRC, 2023.
     3             SRGXT1,MMPIN,KGPHA,METOPT,CCFAC,SCEP1,SRGXT,RAIN,SRC,2024.
     4             SCF,IDNS,F,DGM,WC,MPACK,EVAPSN,MELEV,TSNOW,PETMIN,   2025.
     5             DEWX,DEPTH,MONTH,TMIN,PETMAX,ELDIF,SDEN,WINDX,INFTOM,2026.
     6             TSNBAL,ROBTOM,ROBTOT,RXB,ROITOM,ROITOT,YEAR,CUNIT(7),2027.
     7             INFTOT,MNAM,RAD,SRCI,FORM(42)                        2028.
C                                                                       2029.
      COMMON /QLS/ WSNAME(6),KRER,JRER,KSER,JSER,TEMPCF,COVMAT(5,12),   2030.
     1             KEIM,JEIM,NDSR,ARP(5),ARI(5),ACCP(5),ACCI(5),RPER(5),2031.
     2             PMP(5,5),PMI(5,5),QSNOW,SNOWY,SEDTM,SEDTY,SEDTCA,    2032.
     3             ACPOLP(5,5),ACERSN(5),APOLP(5,5),AERSN(5),COVER(5),  2033.
     4             APOLI(5,5),ACEIM(5),AEIM(5),POLTM(5),POLTY(5),       2034.
     5             TEMPA,DOA,POLTCA(5),AERSNY(5),AEIMY(5),APOLPY(5,5),  2035.
     6             APOLIY(5,5),POLTC(5),PLTCAY(5),ACPOLI(5,5),RIMP(5),
     $             PMD(5,5),STEPS
      COMMON /LNDOUT/ ROSTOM,RINTOM,RITOM,RUTOM,BASTOM,RCHTOM,PRTOM,    2036.
     1             SUMSNM,PXSNM,MELRAM,RADMEM,CONMEM,CDRMEM,            2039.
     2             CRAINM,SGMM,SNEGMM,PACKOT,SEVAPM,EPTOM,NEPTOM,       2040.
     3             UZSOT,LZSOT,SGWOT,SCEPOT,RESSOT,SRGXTO,TWBALO,       2041.
     4             TSNBOL,ROSTOT,RINTOT,RITOT,RUTOT,BASTOT,RCHTOT,      2042.
     5             PRTOT,SUMSNY,PXSNY,MELRAY,RADMEY,CONMEY,CDRMEY,      2043.
     6             CRAINY,SGMY,SNEGMY,PACK1,SEVAPY,EPTOT,NEPTOT,        2044.
     7             UZSMT,LZSMT,SGWMT,SCEPT,RESST,SRGXTT,TWBLMT          2045.
C                                                                       2046.
```

```
      COMMON /STS/ ACPOLT(5),PLTMX(5),POLTSC(5),PLTMXC(5),
     1             ACSEDT,SEDMX,SEDTSC,SEDMXC,TOTRUN,PEAKRU
C
      COMMON /INTM/ RTYPE(4,4),UTYPE(2),GRAD,RADDIS,WINDIS,ICS,OFS,
     1              TEMPAY,DOAY,NOSIY,INTRVL,WMUL,NN,L,SS,NNI,LI,SSI,
     2              RMUL,KUGI,SEDTCY,REPERV(12),REIMPV(12),ACUPV(12),
     3              ACUIV(12),PMPMAT(12,5),PMIMAT(12,5),PMPVEC(5),
     4              PMIVEC(5),ACUI,ACUP,REIMP,REPER,PRINTR,
     5              EPXM(12),TIMTIL(12),SRERTL(12),TILDAY(5,12),
     6              TILSED(5,12),DPM(12),TCF(12),PMDVEC(5),PMDMAT(12,5)
C
C**********
      COMMON/PARM/IDBUG,LDBUG,DOT,EXAMS,ISHED
C
      COMMON/UNITS/NIN,NMET,NOUT,KR,ILP,ILUDOT
C**********
C
      DATA  LAST/.FALSE./, PREV/.FALSE./
      DATA  PRTOT/0.0/
      DATA  PRTOM,PRTM/2*0.0/
      DATA  RUTOM, ROSTOM, RITOM, RINTOM, NEPTOM/5*0.0/
      DATA  RUTOT, ROSTOT, RITOT, RINTOT, NEPTOT/5*0.0/
      DATA  ROBTOM, ROBTOT, INFTOM, INFTOT, ROITOM, ROITOT/6*0.0/
      DATA  TWBAL, RESB, RESBI, ROSBI, RESBI1,SRGX, INTF/7*0.0/
      DATA  RESB1, BASTOM, RCHTOM, BASTOT, RCHTOT/5*0.0/
      DATA  EPTOM, EPTOT/2*0.0/
      DATA  PR, P3, RXB, RGX, RUZB, UZSB, PERCB, DPST/8*0.0/
      DATA  TIMFAC, UZSN, LZSN, INFIL, INTER, IRC/6*0.0/
      DATA  A, UZS, LZS, SGW, GWS, KV, K24L, K24EL, KK24/9*0.0/
      DATA  IFS, K3/2*0.0/
      DATA  EPXM, EPXMI/13*0.0/
      DATA  DPM/31,28,31,30,31,30,31,31,30,31,30,31/
      DATA  TIMTIL, TILDAY/72*0/
      DATA  SRERTL, TILSED/72*0.0/
      DATA  TCF/12*1.0/
      DATA  PETMIN,PETMAX/35.,40./
      DATA  TOTRUN,PEAKRU,ACSEDT,SEDMX,SEDTSC,SEDMXC/6*0.0/
      DATA  ACPOLT,PLTMX,POLTSC,PLTMXC/20*0.0/
      DATA  MNAM/' JAN','UARY',' FEBR','UARY',' MAR','CH  ',' APR',
     *      'IL  ',' MAY','    ',' JUN','E   ',' JUL','Y   ',' AUG',
     *      'UST ','SEPT','MBER',' OCT','OBER','NOVE','MBER','DECE',
     *      'MBER'/
      DATA  MMPIN/25.4/, METOPT/0.9072/, KGPLB/0.4536/, KGPHA/0.892/
      DATA SUMSNM, PXSNM, MELRAM, RADMEM, CDRMEM, CRAINM,PACK,DEPTH,
     *      CONMEM, SGMM, SNEGMM, SEVAPM, SUMSNY, PXSNY, MELRAY,
     *      RADMEY, CDRMEY, CONMEY, CRAINY, SGMY, SNEGMY, SEVAPY,
     *      TSNBAL/23*0.0/
      DATA  INTRVL,PRINTR/15,15/,  WMUL,RMUL,KUGI,SFLAG/1.0,1.0,0.0,0/
      DATA  ICS, OFS/2*0.0/
      DATA GRAD/0.04,0.04,0.03,0.02,
     *0.02,0.02,0.02,0.06,0.14,0.18,0.20,0.17,0.13,0.06,0.03,0.01,0.05,
     *0.07,0.10,0.13,0.15,0.13,0.12,0.08/
      DATA RADDIS/6*0.0,0.019,
     *0.041,0.067,0.088,0.102,0.110,0.110,0.110,0.105,0.095,0.081,0.055,
     *0.017,5*0.0/
      DATA WINDIS/7*0.034,0.035,
     *0.037,0.041,0.046,0.050,0.053,0.054,0.058,0.057,0.056,0.050,0.043,
     *0.040,0.038,0.036,0.036,0.035/
      DATA NN,NNI/.2,.1/, L,LI/2*100./, SS,SSI/2*.01/
      DATA TEMPAY,DOAY,SEDTCA,SEDTCY/4*0.0/, NOSIY,NOSY/2*0/
      DATA FORM/4H(17X , 4H,A4, , 4H4X,A , 4H4,7X , 4H  ,4 , 4HX,'( ,
     *         4HLB)' , 4H,2X, ,4H'(GM , 4H/L)', 4H  ,4 , 4HX,'( ,
     *         4HLB)' , 4H,2X, ,4H'(GM , 4H/L)', 4H  ,4 , 4HX,'( ,
     *         4HLB)' , 4H,2X, ,4H'(GM , 4H/L)', 4H  ,4 , 4HX,'( ,
     *         4HLB)' , 4H,2X, ,4H'(GM , 4H/L)', 4H  ,4 , 4HX,'( ,
     *         4HLB)' , 4H,2X, ,4H'(GM , 4H/L)', 4H  ,4 , 4HX,'( ,
     *         4HLB)' , 4H,2X, ,4H'(GM , 4H/L)', 4H ,2( , 4H/))  /
      DATA ALTR/4H    , 4HKG)',
     *       4H'(MG, 4H( 21, 4H( 27, 4H( 41, 4H( 54, 4H( 63, 4H( 74 /
      DATA TIT/4H    , 4HX,'Q, 4H U A, 4H L I,  4H T Y, 4H   C,
     *    4H O N, 4H S T, 4H I T, 4H U E, 4H N T, 4H S', ,4H / )/
      DATA RTYPE/8H         ,'SEDIMENT','PRODUCTI','   PRODU','  HYDROL',
     *' AND QUA','ON (PRIN','CTION (O','OGIC CAL','LITY CAL','TER OUTP',
     *'UTPUT ON','IBRATION','IBRATION','UT ONLY)',' UNIT 4)'/
      DATA  UTYPE/'  METRIC',' ENGLISH'/
      DATA  COVMAT/60*0.0/, COVER/5*0.0/
      DATA  IMPK/5*0./, NDSR,IHRR/2*0/
      DATA  PMP/25*0.0/, PMI/25*0.0/,PMD/25*0.0/
      DATA QUALIN/' BOD',2*4H    ,' TDS',11*4H    /
      DATA QSNOW/' NO '/, SNOWY/'YES '/
      DATA JRER/0.0/, KRER/0.0/
      DATA JSER/0.0/, KSER/0.0/
      DATA  JEIM/0.0/, KEIM/0.0/
      DATA UNT1/' KG ',' MM ',' LB ',' IN '/
      DATA  STMCH/4800*0.0/
      DATA UNT2/'  T ','CMS ','TONS','CFS '/
      DATA UNT3/'(C) ',' HA ','(F) ','ACRE'/
      DATA AERSN/5*0.0/, AEIM/5*0.0/, APOLP/25*0.0/, APOLI/25*0.0/
      DATA AERSNY/5*0.0/, AEIMY/5*0.0/, APOLPY/25*0.0/, APOLIY/25*0.0/
      DATA TEMPA,DOA/2*0.0/ ,NOSI,NOSIM,NOS/3*0/
      DATA POLTCA/5*0.0/,PLTCAY/5*0.0/
      DATA ACPOLP/25*0.0/, ACPOLI/25*0.0/
      DATA ACEIM,ACERSN/10*0.0/
      DATA  ACCP/5*0./, ACCI/5*0./, RIMP/5*0./, RPER/5*0./
      DATA  SRER/5*0./, TS/5*0./, LMTS/5*0/
      DATA  PMPVEC,PMIVEC,PMPMAT,PMIMAT/5*0.,5*0.,60*0.,60*0./
      DATA PMDVEC/5*0.0/,PMDMAT/60*0.0/
      DATA  ACUP,ACUI,ACUPV,ACUIV/0.,0.,12*0.,12*0./
      DATA  REPER,REIMP,REPERV,REIMPV/0.,0.,12*0.,12*0./
      DATA IDBUG/.FALSE./,LDBUG/.FALSE./
C     DEBUGGING FLAGS
C**********
      DATA DOT/.TRUE./,EXAMS/.TRUE./,ISHED/1/
      DATA NIN/1/,NMET/2/,NOUT/3/,KR/4/,ILP/6/,ILUDOT/7/
C**********
C     LOGICAL UNITS
      END
      SUBROUTINE LANDS
C                               HSP LANDS                               3000.
      IMPLICIT  REAL(L,K)                                               3003.
      LOGICAL LAST, PREV                                                3006.
      LOGICAL IDBUG,LDBUG,NOSNOW                                        3042.
C                                                                       3043.
      INTEGER   TF,HYCAL, DAY, UNIT, SNOW, HRFLAG, H, SFLAG ,LMTS,STRBGN,3044.
     1          RECOUT,YEAR                                             3045.
C                                                                       3046.
      REAL  INFIL, INTER, INFLT, IRC, INTF, INFL
      REAL IFS,NEPTOM,NEPTOT
      REAL  INFTOM, INFTOT, IMPK
      REAL  MMPIN, METOPT, KGPLB
      REAL IDNS, MPACK, MELEV,  NEGMLT, NEGMM                           3050.
      REAL MELT, INDT, KCLD, IPACK, MELRAM, MELRAY                      3054.
      DIMENSION EVDIST(24),LAPSE(24),SVP(40),SNOUT(24,17),STRBGN (4),   3008.
     1          MNAM(24),RAD(24),TEMPX(24),WINDX(24),RAIN(96),DUM1(5),  3009.
     2          DUM2(5)                                                 3010.
C                                                                       3011.
      COMMON /ALL/ RU,HYMIN,HYCAL,DPST,UNIT,TIMFAC,LZS,AREA,RESB,SFLAG, 3012.
     1             RESB1,ROSB,SRGX,INTF,RGX,RUZB,UZSB,PERCB,RIB,P3,TF,  3013.
     2             KGPLB,LAST,PREV,TEMPX,IHR,IHRR,PR,RUI,A,PA,GWF,NOSY, 3014.
     3             SRER(5),TS(5),LNDUSE(3,5),AR(5),QUALIN(3,5),NOSI,NOS,3015.
     4             NOSIM,UFL,UTMP,UNT1(2,2),UNT2(2,2),UNT3(2,2),WHGT,   3016.
     5             WHT,DEPW,ROSBI,RESBI,RESBI1,ARUN,LMTS(5), IMPK(5),   3017.
     6             NLAND,NQUAL,STMCH(200,24),RECOUT(5),FLOUT,SCALEF(5), 3018.
     7             SNOW,PACK,IPACK,SGEN(5)
C                                                                       3020.
      COMMON /LAND/DAY,PRTM,IMIN,IX,TWBAL,SGW,GWS,KV,LIRC4,LKK4,ALTR(9),3021.
     1             UZS,IZ,UZSN,LZSN,INFIL,INTER,SGW1,DEC,DECI,TIT(13),  3022.
     2             K24L,KK24,K24EL,EP,IFS,K3,EPXMI,RESS1,RESS,SCEP,IRC, 3023.
     3             SRGXT1,MMPIN,KGPHA,METOPT,CCFAC,SCEP1,SRGXT,RAIN,SRC,3024.
     4             SCF,IDNS,F,DGM,WC,MPACK,EVAPSN,MELEV,TSNOW,PETMIN,   3025.
     5             DEWX,DEPTH,MONTH,TMIN,PETMAX,ELDIF,SDEN,WINDX,INFTOM,3026.
     6             TSNBAL,ROBTOM,ROBTOT,RXB,ROITOM,ROITOT,YEAR,CUNIT(7),3027.
     7             INFTOT,MNAM,RAD,SRCI,FORM(42)
C                                                                       3028.
      COMMON /LNDOUT/ ROSTOM,RINTOM,RITOM,RUTOM,BASTOM,RCHTOM,PRTOM,    3029.
     1                SUMSNM,PXSNM,MELRAM,RADMEM,CONMEM,CDRMEM,         3030.
     2                CRAINM,SGMM,SNEGMM,PACKOT,SEVAPM,EPTOM,NEPTOM,    3031.
     3                UZSOT,LZSOT,SGWOT,SCEPOT,RESSOT,SRGXTO,TWBALO,    3032.
     4                TSNBOL,ROSTOT,RINTOT,RITOT,RUTOT,BASTOT,RCHTOT,   3033.
     5                PRTOT,SUMSNY,PXSNY,MELRAY,RADMEY,CONMEY,CDRMEY,   3034.
     6                CRAINY,SGMY,SNEGMY,PACK1,SEVAPY,EPTOT,NEPTOT,     3035.
     7                UZSMT,LZSMT,SGWMT,SCEPT,RESST,SRGXTT,TWBLMT       3036.
C                                                                       3037.
      COMMON/PARM/IDBUG,LDBUG,DOT,EXAMS,ISHED                           3038.
      COMMON /STS/ ACPOLT(5),PLTMX(5),POLTSC(5),PLTMXC(5),              3039.
     1             ACSEDT,SEDMX,SEDTSC,SEDMXC,TOTRUN,PEAKRU             3040.
C
      COMMON/UNITS/NIN,NMET,NOUT,NRD,ILP,ILUDOT                         3041.
C
      DATA  PERC, INFLT, SBAS, HRFLAG/0.0,0.0,0.0,0/                    3055.
      DATA  SNET1, SNET, SRCH/3*0.0/                                    3056.
      DATA  ROSINT, REPIN, EPIN1, AETR, KF/5*0.0/
      DATA  EVDIST/6*0.0,0.019,0.041,0.067,0.088,0.102,3*0.11,0.105,    3058.
     C      0.095,0.081,0.055,0.017,5*0.0/                              3059.
                                                                        3060.
```

```
       DATA SVP/10*1.005,1.01,1.01,1.015,1.02,
      *1.03,1.04,1.06,1.08,1.1,1.29,1.66,2.13,2.74,3.49,4.40,5.55,6.87,
      *8.36,10.09,12.19,14.63,17.51,20.86,24.79,29.32,34.61,40.67,47.68,
      *55.71,64.88/
       DATA LAPSE/6*3.5,3.7,4.0,4.1,
      *4.3,4.6,4.7,4.8,4.9,5.0,5.0,4.8,4.6,4.4,4.2,4.0,3.8,3.7,3.6/
       DATA  APR, AEPIN/2*0.0/
       DATA  AROSB, AINTF, AROSIT/3*0.0/
       DATA  ARU, ARUI, AROS, ARGXT, ASNET, ASBAS, ASRCH/7*0.0/
       DATA  SUMSN, INDT, KCLD, PXONSN, SEVAPT, RADME, CDRME, LIQW1,
      *      CONME, CRAIN, NEGMLT, SNEGM, NEGMM, LIQS,  LIQW, XICE,
      *      XLNMLT,SGM, SPX, WBAL, SEVAP/21*0.0/
       DATA  SNOUT/408*0.0/
       DATA  CLDF/-1.0/
C
C                     ZEROING OF VARIABLES
C
       IF(LDBUG) WRITE(ILP,8000) TF,PACK,IPACK,IHR
8000   FORMAT('0*****LANDS TF=',I3,' PACK=',F6.2,' IPACK=',F6.2,
      $' IHR=',I2)
       LZS1 = LZS
       UZS1 = UZS
       NUMI = 0
       DPST = 0.0
       PACK1 = PACK
       LIQW1 = LIQW
       PRR = PR
C
       LNRAT=LZS/LZSN
       D3FV=(2.0*INFIL)/(LNRAT*LNRAT)
       D4F= (TIMFAC/60.)*D3FV
C                                            REDUCE INFILTRATION IF ICE EXISTS
C                                             AT THE BOTTOM OF THE PACK -
C                                           ATTEMPT TO CORRECT FOR FROZEN LAND
       IF (SNOW .LT. 1)  GO TO 20
       D4FX=AMAX1(1.0-XICE,0.1)
       D4F = D4F*D4FX
C
    20 RATIO= INTER*EXP(0.693147*LNRAT)
       IF ((RATIO).LT (1.0)) RATIO=1.0
       D4RA= D4F*RATIO
       H = TF/24
C
C                                          TF IS 1 FOR RAIN DAYS, AND 96
C                                          OR 288 FOR NON-RAIN DAYS
C
       IF(TF.GT.2) GO TO 25
C*****      RAIN DAY     *****
       IF(IHR.LE.0) NOSNOW=.TRUE.
C      START OF DAY--INITIALIZE FLAG
       GO TO 28
25     CONTINUE
C*****      NON-RAIN DAY     *****
       IHRR=0
C      RESET INTERNAL HOUR COUNTER
       NOSNOW=.TRUE.
28     CONTINUE
```

```
C
      DO 1480  III=1,TF                                                 3106.
C*****     START BIG LOOP      *****                                    3107.
      LNRAT = LZS/LZSN
      IF (TF .LT. 2)  GO TO 40                                          3109.
      NUMI = MOD(NUMI + 1,H)                                            3110.
C     NUMI CYCLES FROM 0 TO H-1 FOR NON-RAIN DAYS
   40 SBAS = 0.0
      SRCH = 0.0                                                        3116.
      ROS = 0.0                                                         3117.
      RU = 0.0                                                          3118.
      GWF = 0.0                                                         3119.
      RGXT = 0.0                                                        3120.
      PERC = 0.0                                                        3121.
      INFLT = 0.0                                                       3122.
      RESS = 0.0                                                        3123.
C                                                                       3124.
C   TIMFAC - TIME INTERVAL IN MINUTES                                   3125.
C   L      - LENGTH OF OVERLAND SLOPE                                   3126.
C   NN     - MANNING'S N FOR OVERLAND SLOPE                             3127.
C   A      - IMPERVIOUS AREA                                            3128.
C   PA    - PERVIOUS AREA                                               3129.
C                                                                       3130.
C                                                                       3131.
C                                                                       3132.
C                                                                       3133.
C PR IS INCOMING RAINFALL                                               3134.
C P3 IS RAIN REACHING SURFACE(.00'S INCHES)                             3135.
C P4 IS TOTAL MOISTURE AVAILABLE( IN.)                                  3136.
C RESS  IS OVERLAND FLOW STORAGE( IN.)                                  3137.
C D4F IS 'B' IN OP. MANUAL                                              3138.
C RATIO IS 'C' IN OP. MANUAL                                            3139.
C EP -  DAILY EVAP ( IN.)                                               3140.
C EPHR - HOURLY EVAP                                                    3141.
C EPIN - INTERVAL EVAP                                                  3142.
C EPXX - FACTOR FOR REDUCING EVAP FOR SNOW AND TEMP                     3143.
C                                                                       3144.
C                                                                       3145.
C                                                                       3146.
C                                                                       3147.
C     DETERMINE IF SNOWMELT IS TO BE DONE                               3148.
C                                                                       3149.
   50 HRFLAG=0                                                          3150.
      TEST = IMIN/TIMFAC                                                3151.
      IF (NUMI .EQ. 1)  HRFLAG = 1                                      3152.
      IF ((TEST .LE. 1.001).AND.(TEST .GE. 0.999)) HRFLAG = 1           3153.
C                                                                       3154.
C     HRFLAG=1 INDICATES BEGINNING OF THE HOUR                          3155.
C                                                                       3156.
      IF(HRFLAG.LE.0) GO TO 770                                         3157.
C     SNOWMELT CALCULATIONS DONE ONLY ONCE PER HOUR
   60 IEND = 0
      IF(IHR.EQ.24) GO TO 80                                            3159.
C     IHR = 0-23 FOR RAIN DAYS
   70 IHRR = IHR + 1
      GO TO 90                                                          3161.
80    CONTINUE                                                          3162.
```

```
C     IHR = 24 FOR NON-RAIN DAYS
      IHRR=IHRR+1
C     INCREMENT INTERNAL HOUR COUNTER
90    CONTINUE
      IF(IHRR.GT.24) WRITE(ILP,8004) IHRR,IHR,TF
8004  FORMAT('0*****LANDS ERROR--IHRR=',I2,' EXCEEDS LIMITS',
     $'  IHR=',I2,' TF=',I3)
       EPHR = EVDIST(IHRR)*EP
       IF (EPHR.LE.(0.0001))  EPHR=0.0
       EPIN= EPHR
       EPIN1=EPIN
      IF (SNOW .EQ. 0) GO TO 770
C                                     ***********************************
C                                                BEGIN SNOWMELT
C                                     ***********************************
      TSNOW1 = TSNOW + 1.
      SEVAP = 0.0
      EPXX = 1.0
      IKEND=(60.0/TIMFAC)+0.5
      IPT = (IHRR-1)*IKEND
C                                     SUM PRECIP FOR THE HOUR
      PRHR=0.0
      DO 100 II = 1,IKEND
  100 PRHR = PRHR + RAIN(IPT+II)
C
C                                         CORRECT TEMP FOR ELEVATION DIFF
C                                         USING LAPSE RATE OF 3.5 DURING RAIN
C                                         PERIODS, AND AN HOURLY VARIATION IN
C                                         LAPSE RATE (LAPSE(I)) FOR DRY PERIOD
      LAPS = LAPSE(IHRR)
      IF (PRHR .GT. 0.05)  LAPS = 3.5
      TX = TEMPX(IHRR) - LAPS*ELDIF
C
C
C
C                                      REDUCE REG EVAP FOR SNOWMELT
C                                      CONDITIONS BASED ON PETMIN AND
C                                      PETMAX VALUES
      PACKRA=AMIN1(PACK/IPACK,1.0)
      E1E=1.0 - PACKRA
  130 EPXX = (1.0-F)*E1E + F
      IF(TX.GE.PETMAX) GO TO 170
      EPXX=AMIN1(EPXX,0.5)
C
C
C                                      REDUCE EVAP BY 50% IF TX IS BETWEEN
150   IF(TX.LT.PETMIN) EPXX=0.0        PETMIN AND PETMAX
C
C
  170 EPHR = EPHR*EPXX
      EPIN=EPIN*EPXX
      IEND=0
      IF ((TX .GT. TSNOW) .AND. (PRHR .GT. .02)) DEWX = TX
C
C     SET DEWPT TEMP EQUAL TO AIR TEMP WHEN RAINING
C     ON SNOW TO INCREASE SNOWMELT
C
```

```
      IF (DEWX .GT. TX) DEWX = TX
      SNTEMP = TSNOW + (TX-DEWX)*(0.12 + 0.008*TX)
C
C     RAIN/SNOW TEMP. DIVISION - SEE ANDERSON, WRR, VOL. 4, NO. 1,
C     FEB. 1968, P. 27, EG. 28
C
      SNTEMP=AMIN1(SNTEMP,TSNOW1)
      SFLAG=0
      IF(TX.LT.SNTEMP) SFLAG=1
C     SET FLAG IF TEMP FAVORABLE FOR SNOWFALL
      IF(PACK.GT.0.0.OR.(SFLAG.EQ.1.AND.PRHR.GT.0.0)) IEND=1
C
C     SNOWMELT CALCULATIONS ARE DONE IF IT IS SNOWING, OR,
C     IF A SNOWPACK EXISTS
C
      PX = PRHR
      IF(PX.LE.0.0) GO TO 250
C                                   KCLD IS INDEX TO CLOUD COVER
  210 KCLD = 35.
      IF (SFLAG) 260, 260, 220
C                                  SNOW IS FALLING
  220 PX = PX*SCF
      APR = APR+(SCF-1.0)*PRHR
      PRHR = PRHR*SCF
      SUMSN = SUMSN + PX
      DNS = IDNS
      IF (TX .GT. 0.0)  DNS = DNS + ((TX/100.)**2)
C
C     SNOW DENSITY WITH TEMP. - APPROX TO FIG. 4, PLATE B-1
C     SNOW HYDROLOGY  SEE ALSO ANDERSON, TR 36, P. 21
C
      PACK = PACK + PX
C
      IPACK=AMAX1(IPACK,PACK)
      IPACK=AMIN1(IPACK,MPACK)
C     REDEFINE IPACK
  240 DEPTH = DEPTH + (PX/DNS)
      IF (DEPTH .GT. 0.0) SDEN = PACK/DEPTH
      INDT = INDT - 1000*PX
      IF (INDT .LT. 0.0) INDT = 0.0
      PX = 0.0
      GO TO 260
  250 KCLD = KCLD - 1.
  260 IF (KCLD .LT. 0.0) KCLD = 0.0
      PACKRA = AMIN1(PACK/IPACK,1.0)
C
270   IF(PACK.GE.0.005) GO TO 300
C*****      LAST CHANCE TO AVOID SNOWMELT     *****
C
C     IPACK IS AN INDEX TO AREAL COVERAGE OF THE SNOWPACK
C     FOR INITIAL STORMS IPACK = .1*MPACK SO THAT COMPLETE
C     AREAL COVERAGE RESULTS.  IF EXISTING PACK GT .1 *MPACK THEN
C     IPACK IS SET EQUAL TO MPACK WHICH IS THE WATER EQUI. FOR
C     COMPLETE AREAL COVERAGE PACKRA IS THE FRACTION AREAL COVERAGE
C     AT ANY TIME.
C
  280 IPACK = 0.1*MPACK
```

```
      IF(IDBUG.AND.(.NOT.NOSNOW)) WRITE(ILP,8002) IHR,IHRR,PACK,LIQW,PX,
     $TX
 8002 FORMAT(' *****LANDS--ENDING SNOWMELT IHR=',I2,' IHRR=',I2,
     $' PACK=',F8.2,' LIQW=',F8.2,' PX=',F8.2,' TX=',1PE10.2)
      XICE = 0.0
      XLNMLT = 0.0
      NEGMLT = 0.0
      PX = PX + PACK + LIQW
C     REMAINING PACK AND LIQUID WATER CONSIDERED AS PRECIPITATION
      PACK = 0.0
      LIQW = 0.0
C     RESET VARIABLES
      GO TO 760
  300 PXONSN = PXONSN + PX
      IF (DEPTH .GT. 0.0) SDEN = PACK/DEPTH
      IF (INDT .LT. 800.) INDT = INDT + 1.
C                             INDT IS INDEX TO ALBEDO
      MELT = 0.0
      IF (SDEN .LT. 0.55) DEPTH=DEPTH*(1.0 - 0.00002*(DEPTH*(.55-SDEN)))
C
C     EMPIRICAL RELATIONSHIP FOR SNOW COMPACTION
C
      IF (DEPTH .GT. 0.0) SDEN = PACK/DEPTH
      WIN = WINDX(IHRR)
C
C     HOURLY WIND VALUE
C
      LREF = (TX + 100.)/5
      IREF = IFIX(LREF)
      SVPP = SVP (IREF)
      ITX = IFIX(TX)
      SATVAP = SVPP +(MOD(ITX,5)/5)*(SVP(IREF + 1) - SVPP)
      LREF = (DEWX + 100.)/5
      IREF = IFIX(LREF)
      SVPP = SVP (IREF)
      IDEWX = IFIX(DEWX)
      VAPP = SVPP + (MOD(IDEWX,5)/5)*(SVP(IREF + 1) - SVPP)
C                     CALCULATION OF VAPOR PRESSURE AT AIRTEMP
C                     AND DEWPOINT
      IF (VAPP - 6.108)  320, 320, 310
  310 CNM = 8.59*(VAPP - 6.108)
      GO TO 330
  320 CNM = 0.0
      DUMMY=(VAPP-SATVAP)*PACKRA
      IF (VAPP .LT. SATVAP) SEVAP = EVAPSN*0.0002*WIN*DUMMY
      PACK = PACK + SEVAP
      SEVAPT = SEVAPT - SEVAP
C
C     CONDENSATION - CONVECTION MELT, EQ. T-29B, P.176, SNOW HYDROLOGY
C     CONV - CONVECTION, CONDS - CONDENSATION
C     SEVAP - EVAP FROM SNOW (NEGATIVE VALUE)
C
  330 CNV = 0.0
      IF (TX .GT. 32.) CNV = (TX-32.)*(1.0 - 0.3*(MELEV/10000.))
      CCXC = CCFAC*.00026*WIN
C
C     .00026 = .00629/24, I.E. .00026 IS THE DAILY COEFFICIENT
```

```
C     (FROM SNOW HYDROLOGY) REDUCED TO HOURLY VALUES.
C
      CONV = CNV*CCXC
      CONDS = CNM*CCXC
C                         CLOUD COVER
C                         CLDF IS FRACTION OPEN SKY - MINIMUM VALUE 0.15
      IF ((IHRR.EQ.1).OR.(CLDF.LT.0.0)) CLDF = (1.0 - 0.085*(KCLD/3.5))
C                         ALBEDO
      IF (MONTH - 9)  340, 340, 360
  340 IF (MONTH - 4)  360, 350, 350
  350 ALBEDO = 0.8 - 0.1*(SQRT(INDT/24.))
      IF (ALBEDO .LT. 0.45)  ALBEDO = 0.45
      GO TO 370
  360 ALBEDO = 0.85 - 0.07*(SQRT(INDT/24.0))
      IF (ALBEDO.LT.0.6) ALBEDO=0.6
C                         SHORT WAVE RADIATION-RA - POSITIVE INCOMING
  370 RA = RAD(IHRR)*(1.0 -ALBEDO)*(1.0-F)
C                         LONG WAVE RADIATION - LW - POSITIVE INCOMING
      DEGHR = TX - 32.0
      IF (DEGHR)  390, 390, 380
  380 LW = F* 0.26*DEGHR + (1.0 - F)*(0.2*DEGHR - 6.6)
      GO TO 400
  390 LW = F*0.2*DEGHR + (1.0 - F)*(0.17*DEGHR - 6.6)
C
C                            LW IS A LINEAR APPROX. TO CURVES IN
C                            FIG. 6, PL 5-3, IN SNOW HYDROLOGY.  6.6
C                            IS AVE BACK RADIATION LOST FROM THE SNOWPACK
C                            IN OPEN AREAS, IN LANGLEYS/HR.
C
C                            CLOUD COVER CORRECTION
  400 IF (LW .LT. 0.0)  LW = LW*CLDF
C
C                            RAIN MELT
      RAINM = 0.0
C
C                                     RAINMELT IS OPERATIVE IF IT IS
C                                     RAINING AND TEMP IS ABOVE 32 F
C
      IF ((SFLAG .LT. 1).AND.(TX .GT. 32.))  RAINM = DEGHR*PX/144.
C                            TOTAL MELT
      RM = (LW + RA)/203.2
C                            203.2 LANGLEYS REQUIRED TO PRODUCE I INCH
C                            RUNOFF FROM SNOW AT 32 DEGREES F
      IF (PACK - IPACK )  410, 430, 430
  410 RM = RM*PACKRA
      CONV = CONV*PACKRA
      CONDS = CONDS*PACKRA
      RAINM = RAINM*PACKRA
      IF (IHRR - 6)  430, 420, 430
  420 XLNEM = 0.01*(32.0 - TX)
      IF (XLNEM .GT. XLNMLT) XLNMLT = XLNEM
  430 RADME = RADME + RM
      CDRME = CDRME + CONDS
      CONME = CONME + CONV
      CRAIN = CRAIN + RAINM
      MELT = RM + CONV + CONDS + RAINM
      IF (MELT)  440, 470, 470
```

```
C*****     MELT LT 0     *****
  440 NEGMM = 0.0
      IF (TX .LT. 32.) NEGMM = 0.00695*(PACK/2.0)*(32.0 - TX)
C
C
C                                      HALF OF PACK IS USED TO CALCULATE
C                                      MAXIMUM NEGATIVE MELT
      TP = 32.0 - (NEGMLT/(0.00695*PACK))
C
C     TP IS TEMP OF THE SNOWPACK
C     0.00695 IS IN. MELT/IN. SNOW/DEGREE F
C
      IF (TP - TX)  460, 460, 450
  450 GM = 0.0007*(TP - TX)
      NEGMLT = NEGMLT + GM
      SNEGM = SNEGM + GM
 460  NEGMLT=AMIN1(NEGMLT,NEGMM)
      MELT = 0.0
C
C
C                             MELTING PROCESS BALANCE
  470 PXBY = (1.0 - PACKRA)*PX
C*****     MELT GE 0     *****
      PX = PACKRA*PX
C
C
C     PXBY IS FRACTION OF PRECIP FALLING ON BARE GRCUND
C
      IF (MELT + PX)   650,650,480
C
C     SATISFY NEGMLT FROM PRECIP(RAIN) AND SNOWMELT
C
  480 IF (MELT - NEGMLT) 490, 500, 500
  490 NEGMLT = NEGMLT - MELT
      MELT = 0.0
      GO TO 510
  500 MELT = MELT - NEGMLT
      NEGMLT = 0.0
C
  510 IF (PX - NEGMLT)  520, 530, 530
  520 NEGMLT = NEGMLT - PX
      PACK = PACK + PX
      PX = 0.0
      GO TO 540
  530 PX = PX - NEGMLT
      PACK = PACK + NEGMLT
      NEGMLT = 0.0
C
  540 IF ((PX + MELT) .EQ. 0.0) GO TO  660
C
C     COMPARE SNOWMELT TO EXISTING SNOWPACK AND WATER CONTENT OF
C     THE PACK
C
      IF(MELT.LE.PACK) GO TO 560
C*****     MELT GT PACK     *****
  550 MELT = PACK + LIQW
      DEPTH = 0.0
      PACK = 0.0
      LIQW = 0.0
      INDT = 0.0
      GO TO 590
  560 PACK = PACK - MELT
      IF (SDEN .GT. 0.0)  DEPTH = DEPTH - (MELT/SDEN)
      IF (PACK .GE. (0.9*DEPTH))  DEPTH = 1.11*PACK
      IF (PACK - 0.001)  570, 580, 580
  570 LIQW = LIQW + PACK
      PACK = 0.0
  580 LIQS = WC*PACK
      IF (SDEN .GT. 0.6) LIQS = WC*(3.0 - (3.33)*SDEN)*PACK
      IF (LIQS .LT. 0.0) LIQS = 0.0
C
C     COMPARE AVAILABLE MOISTURE WITH AVAILABLE STORAGE IN SNOWPACK
C     -LIQS
C
  590 IF ((LIQW + MELT + PX) - LIQS)  610, 610, 600
  600 PX = MELT + PX + LIQW - LIQS
      LIQW = LIQS
      GO TO 620
  610 LIQW = LIQW + MELT + PX
      PX = 0.0
  620 IF (PX - XLNMLT)  640, 640, 630
  630 PX = PX - XLNMLT
      PACK = PACK + XLNMLT
      XICE = XICE + XLNMLT
      XLNMLT = 0.0
      GO TO 650
  640 PACK = PACK + PX
      XICE = XICE + PX
      XLNMLT = XLNMLT - PX
      PX = 0.0
  650 IF (XICE .GT. PACK) XICE = PACK
C
C
C                             END MELTING PROCESS BALANCE
C
  660 IF (DEPTH .GT. 0.0) SDEN = PACK/DEPTH
      IF (SDEN .LT. 0.1) SDEN = 0.1
      IF(IHRR.NE.12) GO TO 700
C*****     GROUNDMELT--DONE ONLY ONCE PER DAY     *****
  670 DGMM = DGM
      IF (TP .LT. 5.0)  TP = 5.0
      IF (TP .LT. 32.)  DGMM = DGMM - DGM*.03*(32.0 - TP)
      IF (PACK - DGMM)  690, 690, 680
  680 PX = PX + DGMM
      PACK = PACK - DGMM
      DEPTH = DEPTH - (DGMM/SDEN)
      SGM = SGM + DGMM
      GO TO 700
  690 PX = PACK + PX + LIQW
      SGM = SGM + PACK
      PACK = 0.0
      DEPTH = 0.0
      LIQW = 0.0
      NEGMLT = 0.0
  700 CONTINUE
```

```
      PX = PX + PXBY                                                  3511.
      SPX = SPX + PX                                                  3512.
C                                                                     3513.
C                               MONTHLY SUMS                          3514.
      SUMSNM = SUMSNM + SUMSN                                         3515.
      PXSNM  = PXSNM  + PXONSN                                        3516.
      MELRAM = MELRAM + SPX                                           3517.
      RADMEM = RADMEM + RADME                                         3518.
      CDRMEM = CDRMEM + CDRME                                         3519.
      CONMEM = CONMEM + CONME                                         3520.
      CRAINM = CRAINM + CRAIN                                         3521.
      SGMM   = SGMM   + SGM                                           3522.
      SNEGMM = SNEGMM + SNEGM                                         3523.
      SEVAPM = SEVAPM + SEVAPT                                        3524.
C                                                                     3525.
C                               YEARLY SUMS                           3526.
      SUMSNY = SUMSNY + SUMSN                                         3527.
      PXSNY  = PXSNY  + PXONSN                                        3528.
      MELRAY = MELRAY + SPX                                           3529.
      RADMEY = RADMEY + RADME                                         3530.
      CDRMEY = CDRMEY + CDRME                                         3531.
      CONMEY = CONMEY + CONME                                         3532.
      CRAINY = CRAINY + CRAIN                                         3533.
      SGMY   = SGMY   + SGM                                           3534.
      SNEGMY = SNEGMY + SNEGM                                         3535.
      SEVAPY = SEVAPY + SEVAPT                                        3536.
C                                  ZERO HOURLY VALUES                 3537.
      SUMSN = 0.0                                                     3538.
      PXONSN = 0.0                                                    3539.
      RADME = 0.0                                                     3540.
      CDRME = 0.0                                                     3541.
      CONME = 0.0                                                     3542.
      CRAIN = 0.0                                                     3543.
      SGM = 0.0                                                       3544.
      SNEGM = 0.0                                                     3545.
      SEVAPT = 0.0                                                    3546.
      SPX = 0.0                                                       3547.
C                            SNOWMELT OUTPUT                          3548.
C                                                                     3549.
      SNOUT(IHRR,1) = PACK                                            3550.
      SNOUT(IHRR,2) = DEPTH                                           3551.
      SNOUT(IHRR,3) = PACKRA
      SNOUT(IHRR,4) = ALBEDO                                          3553.
      SNOUT(IHRR,5) = CLDF                                            3554.
      SNOUT(IHRR,6) = NEGMLT                                          3555.
      SNOUT(IHRR,7) = LIQW                                            3556.
      SNOUT(IHRR,8) = TX                                              3557.
      SNOUT(IHRR,9) = RA                                              3558.
      SNOUT(IHRR,10) = LW                                             3559.
      SNOUT(IHRR,11) = PX                                             3560.
      SNOUT(IHRR,12) = MELT                                           3561.
      SNOUT(IHRR,13) = CONV                                           3562.
      SNOUT(IHRR,14) = RAINM                                          3563.
      SNOUT(IHRR,15) = CONDS                                          3564.
      SNOUT(IHRR,16) = RM
      SNOUT(IHRR,17) = XICE
      NOSNOW=.FALSE.
```

```
C     SET FLAG TO INDICATE SNOWMELT TODAY
C                              CORRECT WATER BALANCE FOR SNOWMELT
C                              PACK AND SNOW EVAP
C
C                              PRR IS INCOMING PRECIP
C                              PX  IS MOISTURE TO THE LAND SURFACE
C                              SEVAP IS SNOW EVAP - NEGATIVE
  760 SNBAL = PRHR+SEVAP-PX-PACK+PACK1-LIQW+LIQW1
      IF ((SNBAL.LT.0.0001).AND.(SNBAL.GT.-0.0001))  SNBAL=0.0
      TSNBAL = TSNBAL + SNBAL
C
C
      PACK1 = PACK
      LIQW1 = LIQW
C               **********************************
C                            END SNOWMELT
C               *********************************
C                                              PX IS TOTAL MOISTURE INPUT TO
C                                              THE LAND SURFACE FROM PRECIP
C                                              AND SNOWMELT DURING THE HOUR
C
  770 IF (IEND .GT. 0) PR=PX*TIMFAC/60.
C                                          IEND GT 0 INDICATES SNOWMELT
C                                          OCCURRED DURING THE HOUR
      IF(HYCAL.GT.1.OR.IHRR.NE.24.OR.NOSNOW) GO TO 775
C     HERE TO PRINT SNOWMELT VARIABLES
C     PRINTOUT IF SNOWMELT OCCURED AT ANY TIME DURING THE DAY
      IF(UNIT.LT.1) GO TO 730
C
C  CONVERSION TO METRIC SNOW OUTPUT
C
      DO 720 IHOUR=1,24
      SNOUT(IHOUR,1)=SNOUT(IHOUR,1)*MMPIN
      SNOUT(IHOUR,2)=SNOUT(IHOUR,2)*MMPIN
      SNOUT(IHOUR,6)=SNOUT(IHOUR,6)*MMPIN
      SNOUT(IHOUR,7)=SNOUT(IHOUR,7)*MMPIN
      SNOUT(IHOUR,8)=0.556*(SNOUT(IHOUR,8)-32.0)
C     CONVERT FROM DEF F TO DEG C
        DO 720 ISNOUT=11,17
          SNOUT(IHOUR,ISNOUT) = SNOUT(IHOUR,ISNOUT)*MMPIN
  720   CONTINUE
730   CONTINUE
      WRITE(ILP,4020) MNAM(IZ),MNAM(IX),DAY
      WRITE(ILP,4000)
C
      DO 750 I=1,24
      WRITE(ILP,4010) I,(SNOUT(I,MM),MM=1,17)
      DO 740 MM=1,17
  740 SNOUT(I,MM)=0.0
C     RESET OUTPUT ARRAY
  750 CONTINUE
C
C
 4000 FORMAT('0','HOUR  PACK   DEPTH PACKRA ALBEDO  CLDF  NEGMELT  LIQW
     *  TX     RA   LW    PX   MELT      CONV    RAINM   CONDS      RM',
     $'     ICE')
 4010 FORMAT(' ',I2,2X,F6.1,2X,F6.1,5(1X,F6.3),1X,F6.2,2F5.1,
     *7(1X,F7.4))
 4020 FORMAT ('0',25X,'SNOWMELT OUTPUT FOR',4X,A4,A4,2X,I2)
C
      NOSNOW=.TRUE.
C     RESET 'NO SNOWMELT' FLAG
775   CONTINUE
C
C
C   * * *    INTERCEPTION  FUNC.     * * *
C
C
C EPXMI - MAX. INTERCEPTION STORAGE  FOR THAT DAY
C SCEP - EXISTING INTER. STORAGE
C EPX  - AVAILABLE INTER. STORAGE
C RUI  - IMPERVIOUS RUNOFF DURING INTERVAL
C
C
        EPX=EPXMI-SCEP
        IF(EPX.LT.(0.0001))  EPX=0.0
        IF (PR-EPX)   790,780,780
  780   P3= PR-EPX
        SCEP = SCEP+EPX
        GO TO 800
  790   SCEP = SCEP+PR
        P3=0.0
        RU=0.0
        RUI=0.0
C
C ***    OVERLAND IMPERVIOUS FLOW ROUTING ***
C
C
C RXBI = VOLUME OF IMPERVIOUS OVERLAND FLOW ON SURFACE
C  ROSBI = VOLUME OF OVERLAND IMPERVIOUS FLOW TO STREAM
C  RESBI = VOLUME OF OVERLAND IMPERVIOUS Q REMAINING ON SURFACE
C
C
  800 IF (A) 810,810,820
  810 RUI=0.0
      GO TO 930
  820 RXBI=P3+RESBI
      IF (RXBI-0.001) 830,830,840
  830 RUI=RXBI*A
      RESBI=0.0
      ROSBI=RUI
      GO TO 930
  840 F1= RXBI-(RESBI)
      F3= (RESBI)+ RXBI
      IF (RXBI-(RESBI)) 860,860,850
  850 DE= DECI*((F1)**0.6)
      GO TO 870
  860 DE= (F3)/2.0
  870 IF (F3-(2.0*DE)) 890,890,880
  880 DE=(F3)/2.0
  890 IF ((F3)-(.005)) 900,900,910
  900 ROSBI= 0.0
      GO TO 920
  910     DUMV=(1.0+0.6*(F3/(2.0*DE))**3.)**1.67
```

```
      ROSBI=(TIMFAC/60.)*SRCI*((F3/2.)**1.67)*DUMV
      IF ((ROSBI).GT.(.95*RXBI)) ROSBI=.95*RXBI
  920 RESBI= RXBI-ROSBI
      RUI=ROSBI*A
  930 RU=RUI
C
C
C
C   * * *      INTERCEPTION EVAP      * * *
C
C
940   IF(NUMI.NE.0.OR.IMIN.NE.0.OR.SCEP.LE.0.0) GO TO 1000
  960  IF (SCEP-EPIN)  970,980,980
  970  EPIN = EPIN - SCEP
       SNET = SNET + SCEP
       SCEP = 0.0
       GO TO 1000
  980  SCEP=SCEP-EPIN
  990  SNET=SNET+EPIN
       EPIN = 0.0
C
C
C     ***  INFILTRATION FUNC.  ***
C  P4 IS TOTAL MOISTURE
C SHRD = SURFACE DETENTION AND INTERFLOW
C RXX = SURFACE DETENTION
C RGXX = INTERFLOW  COMPONENT
C  RGX = VOLUME TO INTER. DETEN STOR.
C
C
 1000 P4 = P3 + RESB
      RESB1 = RESB
      IF (P4 - D4F) 1010,1010,1020
 1010 SHRD=(P4**2)/(2.0*D4F)
      GO TO 1030
 1020 SHRD= P4 - 0.5*D4F
      IF (P4 - D4RA) 1030,1030,1040
 1030 RXX = (P4**2)/(2.0*D4RA)
      GO TO 1050
 1040 RXX= P4 - 0.5*D4RA
 1050 RGXX  = SHRD-RXX
C
C
C   ***  UPPER ZONE FUNCTION ***
C
C PRE  - % SURFACE DETENTION TO OVERLAND FLOW
C UZSB -  UPPER ZONE STORAGE
C UZS  - TOTAL UPPER ZONE STORAGE
C RUZB - ADDITION TO U.Z. STORAGE DURING INTERVAL
C
      IF (UZSB.LT.0.0)  UZSB=0.0
      UZRA= UZSB/UZSN
      IF (UZRA.GT.6.0) GO TO 1060
      IF (UZRA.GT.2.0) GO TO 1070
      UZI= 2.0*ABS((UZRA/2.0)-1.0) +1.0
      PRE= (UZRA/2.0)*((1.0/(1.0+UZI))**UZI)
      GO TO 1080
 1060 PRE = 1.0
      GO TO 1080
 1070 UZI= (2.0*ABS(UZRA-2.0))+1.0
      PRE= 1.0-((1.0/(1.0+UZI))**UZI)
 1080 RXB= RXX* PRE
        RGX=RGXX*PRE
        RGXX=0.0
        RUZB=SHRD-RGX-RXB
        UZSB=UZSB+RUZB
C
        RIB = P4 - RXB
C
C
C
C   * * *      UPPER ZONE EVAP    * * *
C
C
C REPIN - ACCUM DAILY EVAP POT. FOR L.Z. AND GRDWATER, I.E
C            PORTION NOT SATISFIED FROM U.Z.
C
C
       IF ((NUMI .EQ. 0).AND.(IMIN .EQ. 0))  GO TO 1090
       GO TO 1150
C
 1090   IF (EPIN.LE.(0.0))  GO TO 1150
          EFFECT=1.0
          IF(UZRA-2.0)  1120,1120,1100
 1100     IF (UZSB-EPIN)  1140,1140,1110
 1110     UZSB=UZSB-EPIN
          RUZB= RUZB-EPIN
          SNET=SNET+PA*EPIN
          GO TO 1150
 1120     EFFECT= 0.5*UZRA
          IF (EFFECT.LT.(0.02))  EFFECT=0.02
          IF (UZSB-EPIN*EFFECT)  1140,1140,1130
 1130     UZSB=UZSB - (EPIN*EFFECT)
          RUZB= RUZB-(EPIN*EFFECT)
          EDIFF= (1.0-EFFECT)*EPIN
          REPIN=REPIN + EDIFF
          EDIFF=0.0
          SNET= SNET + (PA*EPIN*EFFECT)
          GO TO 1150
 1140     EDIFF= EPIN - UZSB
          REPIN= REPIN + EDIFF
          EDIFF=0.0
          SNET= SNET + PA*UZSB
          UZSB=0.0
          RUZB=0.0
C
C
C    * * * *    INTERFLOW FUNCTION * * *
C
C     SRGX - INTERFLOW DETENTION STORAGE
C     INTF - INTERFLOW LEAVING STORAGE
C     SRGXT - TOTAL INTERFLOW STORAGE
C     RGXT  - TOTAL INTERFLOW LEAVING STORAGE DURING INTERVAL
C
```

```
 1150   INTF = LIRC4*SRGX
        SRGX=SRGX+(RGX*PA)-INTF
         RU=RU + INTF
        SRGXT= SRGXT + (RGX*PA-INTF)
        RGXT=RGXT + INTF
C
C ***    OVERLAND PERVIOUS FLOW ROUTING ***
C
C
C RXB = VOLUME TO OVERLAND SURFACE DETENTION
C  ROSB = VOLUME OF OVERLAND FLOW TO STREAM
C  RESB = VOLUME OF OVERLAND Q REMAINING ON SURFACE
C
C
      F1= RXB-(RESB)
      F3= (RESB)+ RXB
      IF (RXB-(RESB)) 1170,1170,1160
 1160 DE= DEC*((F1)**0.6)
      GO TO 1180
 1170 DE= (F3)/2.0
 1180 IF (F3-(2.0*DE)) 1200,1200,1190
 1190 DE=(F3)/2.0
 1200 IF ((F3)-(.005)) 1210,1210,1220
 1210 ROSB= 0.0
      GO TO 1230
 1220    DUMV=(1.0+0.6*(F3/(2.0*DE))**3.)**1.67
      ROSB=(TIMFAC/60.)*SRC*((F3/2.)**1.67)*DUMV
      IF ((ROSB).GT.(.95*RXB)) ROSB=0.95*RXB
 1230 RESB= RXB-ROSB
        ROSB = ROSB*PA
       ROSINT = ROSB + INTF
C
C
C
C         * * *  UPPER ZONE DEPLETION * * *
C
C  DEEPL - DIFFERENCE IN UPPER AND LOWER ZONE RATIOS
C  PERCB - UPPER ZONE DEPLETION
C  PERC  - TOTAL U.Z. DEPLETION
C  INFLT  - TOTAL INFILTRATION
C  ROS  - TOTAL OVERLAND FLOW TO THE STREAM
C
        IF ((NUMI .EQ. 0).AND.(IMIN .EQ. 0))  GO TO 1240
        PERCB = 0.0
        GO TO 1280
C
 1240   DEEPL= ((UZSB/UZSN)-(LZS/LZSN))
        IF (DEEPL-.01)   1280,1280,1250
 1250    PERCB=0.1*INFIL*UZSN*(DEEPL**3)
C
      IF (SNOW .GT. 0) PERCB = PERCB*D4FX
C
         IF (UZSB - PERCB) 1260,1260,1270
 1260    PERCB = 0.0
         GO TO 1280
C
 1270    UZSB=UZSB-PERCB
```

```
              PERC=PERC+PERCB
              RUZB = RUZB - PERCB
   1280       INFL= P4-SHRD
                INFLT=INFLT + INFL
           RESS = RESS + RESB
            UZS= UZS + RUZB
           ROS = ROS + ROSB
           IF (UZS .LE. 0.0001) UZS=0.0
C
C END OF BLOCK LOOP
C
             RU=RU + ROS
           IF(RESS.GE.0.0001) GO TO 1300
   1290      LZS = LZS + RESS
             RESS = 0.0
             RESB = 0.0
   1300    IF(SRGXT.GE.0.0001) GO TO 1320
   1310      LZS = LZS + SRGXT/PA
             SRGXT = 0.0
               SRGX = 0.0
C
C
C    * * * LOWER ZONE AND GROUNDWATER  * * *
C
C  SBAS  - BASE STREAMFLOW
C  SRCH  - SUM OF GRDWATER RECHARGE
C PREL - % OF INFILTRATION AND U.Z. DEPLETION ENTERING L.Z
C  F1A  - GROUNDWATER RECHARGE - IE. PORTION OF INFIL.
C          AND U.Z. DEPLETION ENTERING GRDWATER
C  K24L  - FRACTION OF F1A LOST TO DEEP GRDWATER
C
 1320         LZI=1.5*ABS((LZS/LZSN)-1.0)+1.0
              PREL=(1.0/(1.0+LZI))**LZI
              IF (LZS.LT.LZSN)  PREL=1.0-PREL*LNRAT
              F3= PREL*(INFLT)
               F1A = (1.0-PREL)*INFLT
          IF(NUMI.NE.0.OR.IMIN.NE.0) GO TO 1340
 1330         F3 = F3 + PREL*PERC
              F1A = F1A + (1.0-PREL)*PERC
 1340         LZS= LZS+F3
           F1= F1A*(1.0 - K24L)*PA
           GWF=SGW*LKK4*(1.0 + KV*GWS)
           SBAS= GWF
           RU=RU+GWF
           SRCH= F1A*K24L*PA
            SGW=SGW - GWF + F1
            GWS=GWS + F1
C
C  * * *        GROUNDWATER EVAP   * * *
C
C
C LOS  - EVAP LOST FROM GROUNDWATER
C
C     NOTE: EVAP FROM GRDWATER AND LZ IS CALCULATED ONLY DAILY
C
          IF(HRFLAG.NE.1.OR.IHRR.NE.21) GO TO 1430
 1350       IF (GWS .GT. 0.0001)  GWS = 0.97*GWS
```

```
         LOS= SGW*K24EL*REPIN*PA
         SGW=SGW - LOS
         GWS=GWS - LOS
         SNET= SNET + LOS
         REPIN= REPIN - LOS
         IF (GWS.LT.(0.0))  GWS=0.0
C
C  * * *       LOWER ZONE EVAP  * * *
C
C AETR - EVAP LOST FROM L.Z.
C
C
        IF (REPIN.LT.(0.0001))  GO TO 1420
        LNRAT = LZS/LZSN
         IF (K3-1.0)   1370,1360,1360
 1360    KF=50.0
         GO TO 1380
 1370    KF=0.25/(1.0-K3)
 1380    IF (REPIN - (KF*LNRAT))  1390,1400,1400
 1390    AETR= REPIN*(1.0-(REPIN/(2.0*KF*LNRAT)))
         GO TO 1410
 1400    AETR= 0.5*(KF*LNRAT)
 1410    IF (K3.LT.(0.50))  AETR=AETR*(2.0*K3)
         LZS=LZS - AETR
         SNET= SNET + PA*AETR
         ASNET = ASNET + LOS + PA*AETR
 1420    REPIN = 0.0
 1430    SNETI = SNET - SNET1
C
C
C
C WBAL - WATER BALANCE IN THE INTERVAL
C TWBAL - ACCUMULATED WATER BALANCE
C
C
 1440    WBAL = (LZS-LZS1+UZS-UZS1+RESS-RESS1)*PA+(SNET-SNET1+SGW-SGW1+
     X             SCEP-SCEP1+SRCH+SRGXT-SRGXT1+RU-PR)+(RESBI-RESBI1)*A
 1450   IF(ABS(WBAL).LE.1.0E-4) WBAL=0.0
         TWBAL=TWBAL+WBAL
C
        DPS = F1A*PA
        DPST = DPST + DPS
C
C
C                    RESETTING VARIABLES
C
        LZS1=LZS
        UZS1=UZS
        RESS1=RESS
        SCEP1=SCEP
        SRGXT1=SRGXT
        SGW1=SGW
        SNET1=SNET
        RESBI1=RESBI
         ASBAS = ASBAS + SBAS
         ASRCH = ASRCH + SRCH
        APR = APR + PRR
```

```
           ARU = ARU + RU
           ARUI = ARUI + RUI
           AROS = AROS + ROS
           ARGXT = ARGXT + RGXT
          IF(NUMI.NE.0.OR.IMIN.NE.0) GO TO 1470
 1460      AEPIN = AEPIN + EPIN1
           ASNET = ASNET + SNETI
 1470      AROSB = AROSB + ROSB
           AINTF = AINTF + INTF
           AROSIT = AROSIT + ROSINT
C
 1480 CONTINUE
C*****     END OF BIG LOOP     *****
C
C
C                   CUMULATIVE RECORDS
C
      PRTOM = PRTOM + APR
      EPTOM = EPTOM + AEPIN
      RUTOM = RUTOM + ARU
      ROSTOM = ROSTOM + AROS
      RITOM = RITOM + ARUI
      RINTOM = RINTOM + ARGXT
      NEPTOM = NEPTOM + ASNET
      BASTOM = BASTOM + ASBAS
      RCHTOM = RCHTOM + ASRCH
C
           ROBTOM = ROBTOM + AROSB
           ROBTOT = ROBTOT + AROSB
            INFTOM = INFTOM + AINTF
            INFTOT = INFTOT + AINTF
           ROITOM = ROITOM + AROSIT
           ROITOT = ROITOT + AROSIT
C
      PRTOT = PRTOT + APR
      EPTOT = EPTOT + AEPIN
      RUTOT = RUTOT + ARU
      ROSTOT = ROSTOT + AROS
      RITOT = RITOT + ARUI
      RINTOT =RINTOT + ARGXT
      NEPTOT = NEPTOT + ASNET
      BASTOT = BASTOT + ASBAS
      RCHTOT = RCHTOT + ASRCH
C
C          LOGICAL VARIABLES LAST AND PREV ARE USED TO DETERMINE
C          BEGINNING AND END OF EACH STORM. STORM BEGINS IF RU
C          IS LESS THAN HYMIN IN ONE TIME INTERVAL, AND GREATER IN
C          THE FOLLOWING ONE (PREV=.FALSE. , LAST=.TRUE.). STORM ENDS
C          IF THE OPPOSIT OCCURS (PREV=.TRUE. , LAST=.FALSE.)
C
      RUINCH=RU
      RU = (RU*AREA*43560.)/(TIMFAC*720.)
C     CONVERT RUNOFF TO CUFT/SEC FOR TOTAL WATERSHED
      LAST=RU.GE.HYMIN.AND.TF.LE.2
      IF (LDBUG) WRITE (ILP,1484) RU,ROSB ROSBI,ROSBQ,PA,A
 1484 FORMAT (/,' RU,ROSB,ROSBI = ',3(E11.4,2X),/,
     $       ' PA,A = ',2(2X,E11.4))
```

```
      IF(.NOT.LAST) GO TO 1570
      IF (PREV) GO TO 1550
C
C        COUNT NUMBER OF STORMS AND RECORD TIME OF STORM BEGINNING
C*****   START OF STORM     *****
      NOS=NOS+1
      IF (NOS.EQ.1) WRITE(ILP,4045)
      WRITE(ILP,4050) NOS,MNAM(IZ),MNAM(IX),YEAR
      STRBGN(1)=MNAM(IZ)
      STRBGN(2)=DAY
      STRBGN(3)=IHR
      STRBGN(4)=IMIN
C
C        INITIALIZATION OF VARIABLES FOR STORM SUMMARY
      NCSI=0
      TOTRUN=0.
      PEAKRU=0.
      ACSEDT=0.
      SEDMX=0.
      SEDTSC=0.
      SEDMXC=0.
      DO 1495 I=1,5
      ACPOLT(I)=0.
      PLTMX(I)=0.
      POLTSC(I)=0.
      PLTMXC(I)=0.
 1495 LMTS(I)=0
C
C        PRINT INITIAL CONDITION FOR A NEW STORM
C
      IF (HYCAL.EQ.1) GO TO 1530
      WRITE(ILP,4060) WHT,ARUN
C
C        CALCULATE AND PRINT MEAN ACCUMULATION FOR (1) EACH
C        LAND TYPE USE (WEIGHTED BY % OF PERVIOUS AND IMPERVIOUS
C        AREAS), (2) THE ENTIRE WATERSHED AND THE TOTAL PERVIOUS
C        AND IMPERVIOUS AREAS (WEIGHTED BY % OF VARIOUS LAND TYPE USE)
C
       TEM1=0.0
       TEM2=0.0
       TEM3=0.0
       TEM4=0.0
      TEM8=0.0
      DO 1510 I=1,NLAND
      TEM=(SRER(I)+SGEN(I))*(1-IMPK(I))+TS(I)*IMPK(I)
      WHFUN1=(AR(I)/AREA)*(1-IMPK(I))
      WHFUN2=(AR(I)/AREA)*IMPK(I)
      TEM1=TEM1+SGEN(I)*WHFUN1
      TEM8=TEM8+SRER(I)*WHFUN1
      TEM2=TEM2+TS(I)*WHFUN2
      TEM3=TEM3+WHFUN1
      TEM4=TEM4+WHFUN2
      IF (UNIT.GT.-1) GO TO 1500
      WRITE(ILP,4070) (LNDUSE(IK,I),IK=1,3),TEM,SGEN(I),SRER(I),TS(I)
      GO TO 1510
 1500 TEM5=SRER(I)*2.24
      TEM6=TS(I)*2.24
      TEM7=SGEN(I)*2.24
      TEM=TEM*2.24                                                          4077.
C     CONVERT TONS/ACRE TO KG/HA
      WRITE(ILP,4070) (LNDUSE(IK,I),IK=1,3),TEM,TEM7,TEM5,TEM6
      IF (LMTS(I).EQ.1) WRITE(ILP,4040)                                      4079.
 1510 CONTINUE                                                              4080.
      IF (NLAND.EQ.1) GO TO 1530                                            4081.
      IF (TEM3.GT.0.0) TEM1=TEM1/TEM3                                       4082.
      IF (TEM3.GT.0.0) TEM8=TEM8/TEM3                                       4082.
      IF (TEM4.GT.0.0) TEM2=TEM2/TEM4                                       4084.
      TEM=(TEM1+TEM8)*(1-A)+TEM2*A
      IF (UNIT.LT.1) GO TO 1520                                             4087.
      TEM=TEM*2.24                                                          4088.
      TEM1=TEM1*2.24                                                        4089.
      TEM2=TEM2*2.24                                                        4090.
      TEM8=TEM8*2.24                                                        4090.
C     CONVERT TONS/ACRE TO METRIC TONS/HA
 1520 WRITE(ILP,4080) TEM,TEM1,TEM8,TEM2
 1530 CONTINUE                                                              4092.
      WRITE(ILP,4090)                                                        4093.
      IF (HYCAL.GT.1) GO TO 1540                                            4094.
      WRITE(ILP,4110) UFL                                                    4095.
      GO TO 1550                                                            4096.
 1540 WRITE(ILP,TIT)                                                         4097.
      WRITE(ILP,4100) ((QUALIN(I,J),I=1,3),J=1,NQUAL)                        4098.
      IF (UNIT.EQ.-1) GO TO 1545                                            4099.
 1545 WRITE(ILP,FORM) UFL,UTMP                                               4100.
 1550 QMETRC=RU*.0283                                                       4101.
C                                                                           4102.
C                PRINT DATE,TIME,AND FLOW                                   4103.
C                                                                           4104.
      WRITE(ILP,4130) MNAM(IZ),DAY,IHR,IMIN                                  4105.
      NOSI=NOSI+1                                                           4106.
      FLOUT=RU                                                              4107.
      IF (UNIT.GT.0) FLOUT=QMETRC                                           4108.
      WRITE(ILP,4120) FLOUT                                                  4109.
      RECOUT(4)=IHR                                                         4113.
      RECOUT(5)=IMIN                                                        4114.
C     SAVE TIME FOR FILE OUTPUT
      IF (RU.GT.PEAKRU) PEAKRU=RU                                           4115.
      TOTRUN=TOTRUN+RUINCH                                                  4116.
 1570 APR = 0.0                                                             4117.
      AEPIN = 0.0                                                           4118.
      ARU = 0.0                                                             4119.
      ARUI = 0.0                                                            4120.
      AROS = 0.0                                                            4121.
      ARGXT = 0.0                                                           4122.
      ASNET = 0.0                                                           4123.
      ASBAS = 0.0                                                           4124.
      ASRCH = 0.0                                                           4125.
      AROSB = 0.0
      AINTF = 0.0
      AROSIT = 0.0
      IF (LAST.OR..NOT.PREV) GO TO 1640                                     4129.
C                                                                           4130.
C                     STORM SUMMARY                                         4131.
C                                                                           4132.
```

```
      IF (UNIT.LT.1) GO TO 1590
      TOTRUN=TOTRUN*25.4
      PEAKRU=PEAKRU*0.0283
      ACSEDT=ACSEDT*0.9072
C     CONVERT TONS TO METRIC TONS
      SEDMX=SEDMX*0.454
      DO 1580 I=1,NQUAL
      ACPOLT(I)=ACPOLT(I)*0.454
 1580 PLTMX(I)=PLTMX(I)*0.454
 1590 WRITE(ILP,4150)
      WRITE(ILP,4140) NOS
      WRITE(ILP,4160) NOSI,(STRBGN(I),I=1,4),MNAM(IZ),DAY,IHR,
     *
     *              IMIN,DEPW,TOTRUN,UFL,PEAKRU
      IF (HYCAL.EQ.1) GO TO 1610
      WRITE(ILP,4170) ((QUALIN(I,J),I=1,3),J=1,NQUAL)
      WRITE(ILP,4180) WHT,ACSEDT,(ACPOLT(I),I=1,NQUAL)
      WRITE(ILP,4190) WHGT,SEDMX,(PLTMX(I),I=1,NQUAL)
      SEDTSC=SEDTSC/NOSI
      DO 1600 I=1,5
 1600 POLTSC(I)=POLTSC(I)/NOSI
      DO 1605 I=1,NQUAL
      DUM1(I)=POLTSC(I)/SCALEF(I)
 1605 DUM2(I)=PLTMXC(I)/SCALEF(I)
      WRITE(ILP,4200) SEDTSC,(DUM1(I),I=1,NQUAL)
      WRITE(ILP,4210) SEDMXC,(DUM2(I),I=1,NQUAL)
 1610 WRITE(ILP,4150)
C
C            ACCUMULATION FOR OVERALL STORM SUMMARY
C
      IF (HYCAL.NE.1) GO TO 1620
      STMCH(NOSY+NOS,1)=TOTRUN
      STMCH(NOSY+NOS,2)=PEAKRU
      GO TO 1640
 1620 IF (NOSY+NOS.GT.200) GO TO 1640
      STMCH(NOSY+NOS,1)=ACSEDT
      STMCH(NOSY+NOS,2)=SEDMX
      STMCH(NOSY+NOS,3)=SEDTSC
      STMCH(NOSY+NOS,4)=SEDMXC
      DO 1630 I=1,NQUAL
      II=4*I
      STMCH(NOSY+NOS,II+1)=ACPOLT(I)
      STMCH(NOSY+NOS,II+2)=PLTMX(I)
      STMCH(NOSY+NOS,II+3)=POLTSC(I)
      STMCH(NOSY+NOS,II+4)=PLTMXC(I)
 1630 CONTINUE
      WRITE(ILP,4030)
 1640 CONTINUE
      PREV=LAST
C
C                       FORMAT STATEMENTS
C
 4030 FORMAT ('0')
 4040 FORMAT ('+',70X,'** LIMIT REACHED **')
 4045 FORMAT ('1')
 4050 FORMAT (3(/),130('*'),2(/),55X,'OUTPUT FOR STORM NO.',I3,
     1       ' - ',A4,A4,1X,I4)
 4060 FORMAT (//,1X,'ACCUMULATION OF DEPOSITS ON GROUND AT THE ',
     1        'BEGINNING OF STORM,',A4,'/',A4,
     2        /,3X,'LAND USE',8X,'WEIGHTED MEAN',9X,'PERVIOUS',T59,
     $        'PERVIOUS',T78,'IMPERVIOUS',/,
     $        T42,'GENERATED',T59,'DEPOSITED',/)
 4070 FORMAT (1X,3A4,9X,F8.3,3(12X,F8.3))
 4080 FORMAT (3X,'WEIGHTED MEAN',6X,F8.3,3(12X,F8.3))
 4090 FORMAT (//)
 4100 FORMAT (2X,'DATE    TIME    FLOW    TEMP  DO(PPM)       SEDIMENTS',
     1        5(4X,3A4))
 4110 FORMAT ('   DATE    TIME   FLOW (',A4,')')
 4120 FORMAT ('+',14X,F8.3)
 4130 FORMAT (1X,A4,1X,I2,1X,I2,':',I2)
 4140 FORMAT (/,' SUMMARY FOR STORM # ',I3)
 4150 FORMAT (29(' '))
 4160 FORMAT (/,' NUMBER OF TIME INTERVALS',I4,/,
     1        ' STORM BEGINS',3X,A4,1X,I2,1X,I2,':',I2,/,
     2        ' STORM ENDS',5X,A4,1X,I2,1X,I2,':',I2,/,
     3        ' TOTAL FLOW (',A4,')',1X,F10.3,/,
     4        ' PEAK FLOW (',A4,')',2X,F10.3)
 4170 FORMAT (37X,' SEDIMENTS ',4X,5(4X,3A4))
 4180 FORMAT (' TOTAL WASHOFF (',A4,')',14X,F10.2,5X,5(F14.5,2X))
 4190 FORMAT (' MAX WASHOFF (',A4,'/15MIN)',10X,F10.2,5X,5(F14.5,2X))
 4200 FORMAT (' MEAN CONCENTRATION (GM/L)',9X,F10.2,5X,5(F14.5,2X))
 4210 FORMAT (' MAX CONCENTRATION (GM/L)',10X,F10.2,5X,5(F14.5,2X))
C
      RETURN
      END
      SUBROUTINE QUAL                                                   5000.
C                                                                       5001.
      INTEGER HYCAL,TF,UNIT,RECOUT,SFLAG
C                                                                       5029.
C**********
      LOGICAL IDBUG,LDBUG,DOT,EXAMS
C**********
      REAL  JRER, KRER. JSER, KSER,KEIM,JEIM,IMPK
      REAL*8 WSNAME
C                                                                       5002.
      DIMENSION POLP(5,5),POLI(5,5),EIM(5),POLTLU(5,5),POLT(5),         5003.
     1          TSS(5),RER(5),ERSN(5),SER(5),TEMPX(24)                  5004.
C**********
C     THIS REPLACED WITH /EXOUT/ BELOW
C     DIMENSION TOTROF(5),SEDCON(5),POLCON(5,5)
C**********
C                                                                       5005.
      DIMENSION LIMP(5),LIMI(5)                                         5026.
      COMMON /ALL/ RU,HYMIN,HYCAL,DPST,UNIT,TIMFAC,LZS,AREA,RESB,SFLAG, 5006.
     1             RESB1,ROSB,SRGX,INTF,RGX,RUZB,UZSB,PERCB,RIB,P3,TF,  5007.
     2             KGPLB,LAST,PREV,TEMPX,IHR,IHRR,PR,RUI,A,PA,GWF,NOSY, 5008.
     3             SRER(5),TS(5),LNDUSE(3,5),AR(5),QUALIN(3,5),NOSI,NOS,5009.
     4             NOSIM,UFL,UTMP,UNT1(2,2),UNT2(2,2),UNT3(2,2),WHGT,   5010.
     5             WHT,DEPW,ROSBI,RESBI,RESBI1,ARUN,LMTS(5),IMPK(5),    5011.
     6             NLAND,NQUAL,STMCH(200,24),RECOUT(5),FLOUT,SCALEF(5), 5012.
     7             SNOW,PACK,IPACK,SGEN(5)
C                                                                       5014.
      COMMON /QLS/ WSNAME(6),KRER,JRER,KSER,JSER,TEMPCF,COVMAT(5,12),   5015.
     1             KEIM,JEIM,NDSR,ARP(5),ARI(5),ACCP(5),ACCI(5),RPER(5),5016.
     2             PMP(5,5),PMI(5,5),QSNOW,SNOWY,SEDTM,SEDTY,SEDTCA,    5017.
     3             ACPOLP(5,5),ACERSN(5),APOLP(5,5),AERSN(5),COVER(5),  5018.
     4             APOLI(5,5),ACEIM(5),AEIM(5),POLTM(5),POLTY(5),       5019.
     5             TEMPA,DOA,POLTCA(5),AERSNY(5),AEIMY(5),APOLPY(5,5),  5020.
     6             APOLIY(5,5),POLTC(5),PLTCAY(5),ACPOLI(5,5),RIMP(5),  5021.
     $             PMD(5,5),STEPS
      COMMON /STS/ ACPOLT(5),PLTMX(5),POLTSC(5),PLTMXC(5),              5023.
     1             ACSEDT,SEDMX,SEDTSC,SEDMXC,TOTRUN,PEAKRU             5024.
      COMMON/UNITS/NIN,NMET,NOUT,KR,ILP,:LUDOT
C**********
      COMMON/EXOUT/ WETTMP(5),TOTROF(5),SEDCON(5),POLCON(5,5)
      COMMON/PARM/IDBUG,LDBUG,DOT,EXAMS,ISHED
C**********
C                                                                       5025.
      DO 10 I=1,5                                                       5031.
      LIMP(I)=.0                                                        5032.
   10 LIMI(I)=.0                                                        5033.
C                                                                       5034.
C     COMPUTE ACCUMULATION OF SEDIMENTS
      DO 15 I=1,NLAND
      RIMPFR=(1.0-RIMP(I))**(1.0/STEPS)
      ACCIFR=(ACCI(I)+.0005*WETTMP(1))/STEPS
      TS(I)=TS(I)*RIMPFR+ACCIFR
      IF(RIMP(I).LE.0.0) GO TO 12
C     CHECK FOR ASYMPTOTIC ACCUMULATION
      RATIO=ACCI(I)/RIMP(I)
      IF(TS(I).LE.RATIO) GO TO 12
      TS(I)=RATIO
      LMTS(I)=1
12    RPERFR=(1.0-RPER(I))**(1.0/STEPS)
      ACCPFR=(ACCP(I)+.0005*WETTMP(I))/STEPS
      SRER(I)=SRER(I)*RPERFR+ACCPFR
      IF(RPER(I).LE.0.0) GO TO 14
      PRATIO=ACCP(I)/RPER(I)
      IF(SRER(I).LE.PRATIO) GO TO 14
      SRER(I)=PRATIO
14    SGEN(I)=SGEN(I)*RPERFR
15    CONTINUE
      IF(TF.GE.2) RETURN
C     FOR NON-RAIN DAYS ONLY ACCUMULATION CALCULATIONS ARE PERFORMED
C                                                                      5037.
C          CONVERT ROSB - VOLUME OF OVERLAND FLOW REACHING STREAM -    5038.
C                         IN INCHES PER WHOLE WATERSHED TO INCHES      5039.
C                         PER PERVIOUS AREAS ONLY                      5040.
C
      ROSBQ=0.0
      IF(PA.GT.0.0) ROSBQ=ROSB/PA
      CONV=43560.0/(12.0*TIMFAC*60.0)
C     CONVERSION FACTOR--INCHES RUNOFF PER TIME INT TO CUFT/SEC
      IF(RU.LE.0.0) RETURN
C     CHECK MAGNITUDE OF RUNOFF
      DO 90 I=1,NLAND                                                  5047.
      TOTROF(I)=CONV*AR(I)*(ROSBI*IMPK(I)+ROSBQ*(1.0-IMPK(I)))
C**********
      IF (TOTROF(I).EQ.0.) TOTROF(I)=.001
C**********
C     COMPUTE TOTAL RUNOFF FROM AREA (CUFT/SEC)
C                                                                      5048.
C                   IF RAIN ON SNOW, INCREASE COVER BY % OF SNOW COVER 5049.
C                                                                      5050.
      IF (SNOW.EQ.0) GO TO 35
      IF (PACK/IPACK.LT.COVER(I)) GO TO 35
      CR=COVER(I)+(1-COVER(I))*(PACK/IPACK)                            5052.
      IF (CR.LT.COVER(I)) GO TO 35                                     5053.
      IF (CR.LE.1.0) COVER(I)=CR                                       5054.
   35 CONTINUE                                                         5055.
C                                                                      5056.
C                        WASHOFF FROM PERVIOUS AREAS                   5057.
C                                                                      5058.
      IF (SFLAG.EQ.1) GO TO 40                                         5059.
C                                                                      5060.
C                   IF SNOWS, BRANCH OVER FINES GENERATION             5061.
C                                                                      5062.
      RER(I)=(1-COVER(I))*KRER*PR**JRER                                5063.
      SGEN(I)=SGEN(I)+RER(I)                                           5064.
40    CONTINUE
      IF((ROSBQ+RESB).GT.0.0) GO TO 60                                 5066.
      ERSN(I)=0.0                                                      5067.
      DO 50 J=1,NQUAL                                                  5068.
   50 POLP(I,J)=0.0                                                    5069.
      GO TO 90                                                         5070.
   60 SER(I)=KSER*(ROSBQ+RESB)**JSER                                   5071.
      STOT=SRER(I)+SGEN(I)
C     COMPUTE TOTAL AVAILABLE SEDIMENT
      IF(SER(I).LE.STOT) GO TO 70
C     CHECK IF TRANSPORT LIMITED (SER LE STOT) OR
```

```
C      SEDIMENT LIMITED (SER GT STOT)
      SER(I)=STOT
      LIMP(I)=1
   70 ERSN(I)=SER(I)*(ROSBQ/(ROSBQ+RESB))
      RFRAC=0.0
      IF(STOT.GT.0.0) RFRAC=SRER(I)/STOT
C     COMPUTE FRACTION OF DEPOSITED SEDIMENT
      SRER(I)=SRER(I)-ERSN(I)*RFRAC
      SGEN(I)=SGEN(I)-ERSN(I)*(1.0-RFRAC)
      ERSN(I)=ERSN(I)*ARP(I)
      SRER(I)=AMAX1(SRER(I),0.0)
      SGEN(I)=AMAX1(SGEN(I),0.0)
C
C      MONTHLY ACCUMULATION OF WASHOFF FROM PERVIOUS AREAS
C
      DO 80 J=1,NQUAL
      AVGPOT=(PMP(J,I)*(1.0-RFRAC)+PMD(J,I)*RFRAC)/100.0
C     COMPUTE WEIGHTED AVERAGE POTENCY FRACTION
      POLP(I,J)=ERSN(I)*2000.0*AVGPOT
      ACPOLP(I,J)=ACPOLP(I,J)+POLP(I,J)
   80 APOLP(I,J)=APOLP(I,J)+POLP(I,J)
      ACERSN(I)=ACERSN(I)+ERSN(I)
      AERSN(I)=AERSN(I)+ERSN(I)
   90 CONTINUE
C
C                                   WASHOFF FROM IMPERVIOUS AREAS
C
      DO 140 I=1,NLAND
      IF ((ROSBI+RESBI).GT.0.) GO TO 110
      EIM(I)=0.0
      DO 100 J=1,NQUAL
  100 POLI(I,J)=0.0
      GO TO 140
  110 TSS(I)=KEIM*((ROSBI+RESBI)**JEIM)
      IF(TSS(I).LE.TS(I)) GO TO 120
C     CHECK IF TRANSPORT LIMITED
      TSS(I)=TS(I)
      LIMI(I)=1
  120 EIM(I)=TSS(I)*(ROSBI/(ROSBI+RESBI))
      TS(I)=TS(I)-EIM(I)
      EIM(I)=EIM(I)*ARI(I)
      DO 130 J=1,NQUAL
      POLI(I,J)=EIM(I)*(PMI(J,I)/100.)*2000.
      APOLI(I,J)=APOLI(I,J)+POLI(I,J)
  130 ACPOLI(I,J)=ACPOLI(I,J)+POLI(I,J)
      ACEIM(I)=ACEIM(I)+EIM(I)
      AEIM(I)=AEIM(I)+EIM(I)
  140 CONTINUE
C
C                                   STORMWATER TEMPERATURE AND DISSOLVED OXYGEN
C                                                               (ASCE,SE4(86),P41)
C
      TEMPC=(TEMPX(IHRR)*TEMPCF-32.)*5/9
      TEMPC=AMAX1(TEMPC,0.0)
      DO=14.652-0.41022*TEMPC+0.007991*(TEMPC**2)-.000077774*(TEMPC**3)
C
C                                 WASHOFF SUMMARY FOR A GIVEN TIME INTERVAL
```

```
C
      DO 160 J=1,NQUAL                                                  5121.
      POLT(J)=0.000                                                     5122.
      DO 150 I=1,NLAND                                                  5123.
      POLTLU(I,J)=POLP(I,J)+POLI(I,J)                                   5124.
      CFACT=454.0/(TOTROF(I)*TIMFAC*60.0*28.32)                         5125.
C     CONVERSON FACTOR (LBS/TIME INT TO G/L)
      POLCON(I,J)=POLTLU(I,J)*SCALEF(J)*CFACT
      POLT(J)=POLT(J)+POLTLU(I,J)
  150 CONTINUE
      ACPOLT(J)=ACPOLT(J)+POLT(J)/2000.                                 5126.
      PLTMX(J)=AMAX1(PLTMX(J),POLT(J))                                  5127.
      POLTC(J)=POLT(J)*454.*SCALEF(J)/(RU*TIMFAC*60.0*28.32)            5128.
      POLTSC(J)=POLTSC(J)+POLTC(J)
      PLTMXC(J)=AMAX1(PLTMXC(J),POLTC(J))                               5130.
      POLTCA(J)=POLTCA(J)+POLTC(J)                                      5131.
  160 CONTINUE
      SEDT=0.000                                                        5133.
      DO 170 I=1,NLAND                                                  5134.
      SEDT=SEDT+ERSN(I)+EIM(I)                                          5135.
      CFACT=454.0/(TOTROF(I)*TIMFAC*60.0*28.32)                         5136.
C     CONVERSION FACTOR: TOTAL LBS TO G/L                               5137.
      SEDCON(I)=(ERSN(I)+EIM(I))*2000.0*CFACT
  170 CONTINUE
      ACSEDT=ACSEDT+SEDT                                                5138.
      SEDT=SEDT*2000.                                                   5139.
      SEDMX=AMAX1(SEDMX,SEDT)                                           5140.
      SEDTC=SEDT*454./(RU*TIMFAC*60.0*28.32)
      SEDTSC=SEDTSC+SEDTC                                               5142.
      SEDMXC=AMAX1(SEDMXC,SEDTC)                                        5143.
C
C                                                                       5145.
C                      PRINTING OF OUTPUT FOR ONE TIME INTERVAL         5146.
      TEMP=TEMPX(IHRR)*TEMPCF                                           5147.
      TEMP=AMAX1(TEMP,32.0)                                             5148.
      DOA=DOA+DO
      TEMPA=TEMPA+TEMP                                                  5150.
      SEDTCA=SEDTCA+SEDTC                                               5151.
      NOSIM=NOSIM+1                                                     5152.
      IF (RU.LT.HYMIN) RETURN                                           5152.5
C*****     STORM IN PROGRESS     *****
      IF(UNIT.EQ.-1) GO TO 190                                          5154.
      TEMP=TEMPC                                                        5155.
      SEDT=SEDT*0.454                                                   5156.
      DO 180 J=1,NQUAL                                                  5157.
  180 POLT(J)=POLT(J)*0.454                                             5158.
  190 CONTINUE                                                          5159.
      WRITE(ILP,4000) TEMP,DO,SEDT,SEDTC,(POLT(J),POLTC(J),J=1,NQUAL)   5160.
C**********
      IF ((HYCAL.NE.4).OR.(EXAMS)) GO TO 200
C**********
C*****     FILE OUTPUT     *****
      DO 195 I=1,NLAND
      TROUT=TOTROF(I)
      IF(UNIT.EQ.1) TROUT=TOTROF(I)*0.0283
  195 WRITE(NOUT,4100) RECOUT,I,TROUT,TEMP,DO,SEDCON(I),
     $(POLCON(I,J),J=1,NQUAL)
```

```
4100  FORMAT(612,F8.3,2F5.2,6F8.3)
C
C         PRINT OF ADDITIONAL OUTPUT FOR CALIBRATION RUN                5164.
C                                                                       5165.
  200 IF ((SEDT.LE.0.001).OR.(HYCAL.GT.2)) RETURN                      5166.
C     HERE FOR CALIBRATION RUNS (HYCAL = 1 OR 2)
      RUI=RUI*CONV
      TEM=RU
      IF (UNIT.LT.1) GO TO 210                                          5169.
      TEM=TEM*0.0283                                                    5170.
      RUI=RUI*0.0283                                                    5171.
      PR=PR*25.4                                                        5172.
  210 WRITE(ILP,4010) TEM,UFL                                           5173.
      WRITE(ILP,4020) RUI,UFL                                             5174.
      WRITE(ILP,4030) PR,DEPW                                            5175.
      DO 240 I=1,NLAND
      ERSN(I)=ERSN(I)*2000.                                             5177.
      EIM(I)=EIM(I)*2000.                                               5178.
      TEM=ERSN(I)+EIM(I)                                                5179.
      IF (TEM.LE.0.001) GO TO 240                                       5180.
      IF (UNIT.LT.1) GO TO 230                                          5181.
      TEM=TEM*0.454                                                     5182.
      EIM(I)=EIM(I)*0.454                                               5183.
      ERSN(I)=ERSN(I)*0.454                                             5184.
      DO 220 J=1,NQUAL                                                  5185.
      POLTLU(I,J)=POLTLU(I,J)*0.454                                     5186.
      POLP(I.J)=POLP(I,J)*0.454                                         5187.
  220 POLI(I,J)=POLI(I,J)*0.454                                         5188.
  230 WRITE(ILP,4040) (LNDUSE(KK,I),KK=1,3),TEM,(POLTLU(I,J),J=1,NQUAL)  5189.
      IF (LIMP(I).EQ.0)                                                   5190.
     $         WRITE(ILP,4050) COVER(I),ERSN(I),(POLP(I,J),J=1,NQUAL)
      IF (LIMP(I).EQ.1)
     $         WRITE(ILP,4060) COVER(I),ERSN(I),(POLP(I,J),J=1,NQUAL)   5193.
C     LIMP = 1 IF SEDIMENT LIMITED
      IF (LIMI(I).EQ.0) WRITE(ILP,4070) EIM(I),(POLI(I,J),J=1,NQUAL)      5195.
      IF (LIMI(I).EQ.1) WRITE(ILP,4080) EIM(I),(POLI(I,J),J=1,NQUAL)      5196.
  240 CONTINUE
      WRITE(ILP,4090)                                                   5197.
      RETURN
 4000 FORMAT ('+',22X,F6.2,2X,F5.2,F9.2,F8.3,5(F8.2,F8.3))
 4010 FORMAT ('0',3X,'TOTAL FLOW',F8.3,' ',A4)                          5214.
 4020 FORMAT (' ',1X,'IMPERV. FLOW',F8.3,' ',A4)                        5215.
 4030 FORMAT (' PRECIPITATION ',F7.3,' ',A4)                            5216.
 4040 FORMAT ('0',21X,3A4,1X,F10.2,8X,5(F10.3,6X))                      5217.
 4050 FORMAT (' ',8X,'COVER=',F5.2,7X,'PERV.',3X,F10.2,8X,5(F10.3,6X))  5218.
 4060 FORMAT (9X,'COVER=',F5.2,7X,'PERV.',2X,'*',F10.2,8X,5(F10.3,6X))  5219.
 4070 FORMAT (27X,'IMPERV.',1X,F10.2,8X,5(F10.3,6X))                    5220.
 4080 FORMAT (27X,'IMPERV.','*',F10.2,8X,5(F10.3,6X))                   5221.
 4090 FORMAT (/)                                                        5222.
      END                                                               5223.
      SUBROUTINE TDIST(TEMPX,GRAD,MAXPRV,MINTMP,MAXTMP,MINNXT)
C     ROUTINE TO COMPUTE 24-HOUR TEMPERATURE DISTRIBUTION
C     BASED ON MIN AND MAX TEMPS
      DIMENSION TEMPX(24),GRAD(24)
C     MAXPRV--PREVIOUS DAY'S MAXIMUM TEMP
C     (MINTMP,MAXTMP)--CURRENT DAY'S MIN AND MAX
C     MINNXT--NEXT DAY'S MIN TEMP
```

```
      C     DAILY MIN ASSUMED TO OCCUR AT 0600 HOURS, MAX AT 1600
            TSUM=MAXPRV
            DELTMP=MINTMP-MAXPRV
            DO 10 IH=17,24
      10    TSUM=TSUM+DELTMP*GRAD(IH)
            DO 20 IH=1,6
            TSUM=TSUM+DELTMP*GRAD(IH)
      20    TEMPX(IH)=TSUM
      C     COMPUTE TEMPERATURES FROM 0100 TO 0600
            DELTMP=MAXTMP-MINTMP
            TSUM=MINTMP
            DO 30 IH=7,16
            TSUM=TSUM+DELTMP*GRAD(IH)
      30    TEMPX(IH)=TSUM
      C     COMPUTE TEMPERATURES FROM 0700 TO 1600
            DELTMP=MINNXT-MAXTMP
            TSUM=MAXTMP
            DO 40 IH=17,24
            TSUM=TSUM+DELTMP*GRAD(IH)
      40    TEMPX(IH)=TSUM
      C     COMPUTE TEMPERATURES FROM 1700 TO 2400
            RETURN
            END
```

```
//EPATOB   JOB (SW01WQMDDADL,MDMC),'ALWAS RUN',MSGLEVEL=(1,1),
// PRTY=4,TIME=5,REGION=600K
//*PDP11: NAME=PUNCH;UIC=205,204
//CVT EXEC FTG1CG
//FORT.SYSIN DD *
C
C       PURPOSE: TO CONVERT THE NPS CATCHMENT FILES THAT CONTAIN
C                SEVERAL PHYSCIAL RECORDS PER LOGICAL RECORD TO
C                A SINGLE RECORD.
C
C       FILE: DB0: 250,210 CVT0353.FTN
C
C       DEVELOPER, DATE: D. CLINE, 6-AUG-81
C
        REAL INPUT(31),INPUT1(24),INPUT2(2)
        INTEGER*4 K
        EQUIVALENCE(INPUT(1),INPUT1(1),INPUT2(1))
        DATA KNT/0/
        WRITE(6,10)
10      FORMAT('OFILE: NPSFOR007.DAT OPENED FOR INPUT.',/)
        WRITE(6,11)
11      FORMAT('OFILE: FT07F001 OPENED FOR OUTPUT.',/)
100     CONTINUE
        READ(5,1,END=500)I,J,K,N
1       FORMAT(I2,I4,I6,I2)
        IF(N.EQ.2)READ(5,2,END=500)INPUT2
        IF(N.EQ.24)READ(5,2,END=500)INPUT1
        IF(N.EQ.31)READ(5,2,END=500)INPUT
        WRITE(2,3)I,J,K,INPUT
2       FORMAT(7F11.0)
3       FORMAT(I2,I4,I6,31E11.4)
        KNT=KNT+1
        GO TO 100
500     CONTINUE
        WRITE(6,13)KNT
13      FORMAT('ORECORDS PROCESSED: ',I5)
        END
//GO.FT02F001 DD DSN=&&NPSOUT,DISP=(,PASS),
//  UNIT=SYSDA,DCB=(LRECL=353,RECFM=FB,BLKSIZE=7060),
//  SPACE=(TRK,(5,5),RLSE)
//GO.SYSIN DD *
 1  72      131
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00
 1  72      231
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00
 1  72      331
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
```

NPSEXAMS

```
C       NPS - EXAMS INTERFACING ROUTINE
C
C
C  THIS PROGRAM REQUIRES 'NSHED' INPUT FILES FROM NPS, WHERE
C
C      NSHED = NUMBER OF NPS WATERSHEDS
C
C  AS ONE CAN SEE FROM THE 'DO 30' LOOP, THE LOGICAL UNIT NUMBERS
C  MUST BE ARRANGED SO THAT UNIT NUMBERS 9,10,11,... CORRESPOND TO
C  TOXICANT OUTPUT FILES FOR WATERSHEDS 1,2,3,...
C
C                              ROBERT MCARTHUR
C                              ARTHUR D. LITTLE, INC.
C                              CAMBRIDGE, MASS.      (617) 864-5770
C
C                              MARCH, 1981
C
      REAL NPSRO
      INTEGER REC
      LOGICAL DEBUG
      DIMENSION NPSRO(10),REC(3),MAP(10),SCALE(10)
      DIMENSION SEDEX(10),FLOWEX(10)
C
C  ---MAP(I) IS THE EXAMS WATER BODY WHICH RECEIVES RUNOFF FROM
C     NPS WATERSHED/CATCHMENT I.
C  ---SCALE(I) IS THE RATIO (CATCHMENT AREA)/(WATERSHED AREA)
C     FOR NPS WATERSHED I.
C
      DEBUG=.FALSE.
      LUEX1=7
      LUEX2=8
C
      READ (4,1000) NDAY,NSHED,MAP
      READ(4,1001) SCALE
      IF (DEBUG) WRITE (6,5000) NDAY,NSHED,MAP
      IF (DEBUG) WRITE (6,5001) SCALE
C
      DO 50 IDAY=1,NDAY
C
      IF (DEBUG) WRITE (6,8001) IDAY
      DO 20 J=1,10
      SEDEX(J)=0.
      FLOWEX(J)=0.
   20 NPSRO(J)=0.
C
      DO 30 ISHED=1,NSHED
C
      IMAP=MAP(ISHED)
      NPSOUT=8+ISHED
      AREA=SCALE(ISHED)
      IF (DEBUG) WRITE (6,8000) ISHED,IMAP,NPSOUT,AREA
      READ (NPSOUT,2000,END=999) REC,SED,POL,FLOW
      IF (DEBUG) WRITE (6,6000) REC,SED,POL,FLOW
      SEDEX(IMAP)=SEDEX(IMAP)+SED*AREA
      FLOWEX(IMAP)=FLOWEX(IMAP)+FLOW
      NPSRO(IMAP)=NPSRO(IMAP)+POL*AREA
C
```

```
C
   30 CONTINUE
C
C------
C WRITE STATEMENTS MOVED FROM B 4 30 CONTINUE TO HERE
C PER BILL TUCKER, 18-FEB-82
C------
C
      WRITE (LUEX1,3001) NPSRO
      WRITE (LUEX2,3001) SEDEX
      WRITE (LUEX2,3001) FLOWEX
      WRITE (6,4001) NPSRO
      WRITE (6,4002) SEDEX
      WRITE (6,4003) FLOWEX
C
   50 CONTINUE
C
  999 STOP
C
 1000 FORMAT (I4,11I2)
 1001 FORMAT (10(F6.0,1X))
 2000 FORMAT (3I2,4E11.4)
 3001 FORMAT (10E11.4)
 4001 FORMAT (' TOX RUNOFF (KG)',/,'    ',5E11.4,/,'    ',5E11.4)
 4002 FORMAT (' SED RUNOFF (KG)',/,'    ',5E11.4,/,'    ',5E11.4)
 4003 FORMAT (' FLOW (CUBIC METERS)',/,'    ',5E11.4,/,'    ',5E11.4)
 5000 FORMAT (' NO. OF DAYS = ',I4,5X,'NO. OF WATERSHEDS = ',I2,/,
     $        ' WATERSHED-EXAMS CELL MAP = ',10I2)
 5001 FORMAT (' CATCHMENT/WATERSHED AREAS =',/,' ',10(F6.2,2X))
 6000 FORMAT (4X,'Y/M/D = ',3I3,/,5X,'SED = ',E11.4,
     $        /,5X,'POL = ',E11.4,/,5X,'FLOW =  ',E11.4)
 8000 FORMAT (/' WATERSHED NUMBER ',I2,/,
     $        '     ASSOC. EXAMS CELL = ',I2,/,
     $        '     OUTPUT FILE NO. = ',I2,/,
     $        '     C/W AREA RATIO = ',F6.2)
 8001 FORMAT (//,'         *** DAY NUMBER',I4,'***'//)
C
      END
```

EXAMS

```
C >>>>>>....<<<<<  MAIN ENTRY POINT OF 'ADLEXAMS' IS ROUTINE: 'DATAIN'..ADL00010
C                                                                       ADL00020
C                                                                       ADL00030
C     +----------------------------------------------------------------+ADL00040
C     |                                                                |ADL00050
C     |  A D L E X A M S : A MODIFIED VERSION OF 'EXAMS' WHICH COUPLES |ADL00060
C     |  ==================  --------             -----                |ADL00070
C     |                      TO THE OTHER COMPONENTS OF THE A L W A S  |ADL00080
C     |                                                     =========  |ADL00090
C     |                      (AIR-LAND-WATER ANALYSIS SYSTEM) PACKAGE. |ADL00100
C     |                       =   =    =     =        =                |ADL00110
C     |                      THESE MODIFICATIONS WERE MADE BY:         |ADL00120
C     |                           ARTHUR D. LITTLE, INC.               |ADL00130
C     |                           25 ACORN PARK                        |ADL00140
C     |(REL.1-02, 12/82)          CAMBRIDGE, MA. 02140                 |ADL00150
C     +----------------------------------------------------------------+ADL00160
C                                                                       ADL00170
C FILE 'DATAIN.FTN'                                                     ADL00180
C THIS ROOT SEGMENT READS IN THE DATA ON THE TOXICANT, THE              ADL00190
C ECOSYSTEM, AND THE RUN PARAMETERS.  IN THE EXAMS SYSTEM PER SE,       ADL00200
C IT IS REPLACED BY THE USER-INTERACTIVE ROUTINES.                      ADL00210
C----------------------------------------------------------------------  ADL00220
C SUBROUTINES USED BY INTEGRATION ROUTINE:                              ADL00230
        EXTERNAL FCT,OUTP                                               ADL00240
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL00250
        REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG  ADL00260
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL00270
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL00280
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG     ADL00290
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL00300
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL00310
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL00320
     1,NPSEDG,INTFLG                                                    ADL00330
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH     ADL00340
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL00350
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL00360
C**********                                                             ADL00370
      LOGICAL NPS,DOT                                                   ADL00380
      COMMON/NPSBLK/NPS,LUNPS,LUNPS2,NPSPER                             ADL00390
      COMMON/DOTBLK/DOT,LUDOT                                           ADL00400
C**********                                                             ADL00410
      COMMON/NAME1G/CHEMNA(60)                                          ADL00420
      COMMON/CONTRG/SPFLG(5)                                            ADL00430
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL00440
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL00450
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL00460
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL00470
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL00480
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL00490
     1,KBHG(3,5),EBHG(3,5)                                              ADL00500
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL00510
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL00520
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL00530
     1,IFLLDG(10)                                                       ADL00540
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL00550
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL00560
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL00570
```

```
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG
     1,DFACG(10)
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)
     1,INTFLG(10)
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:
C-----------------------------------------------------------------------
C FILE 'CHEML.COM'
       COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)
       COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),
     2 VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)
       COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)
       COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV
       COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),
     2 CHEMPC,BIOPCT,VOLPCT,EXPPCT
       COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)
       REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L
C-----------------------------------------------------------------------
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:
C-----------------------------------------------------------------------
C FILE 'ENVIRL.COM'
       COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)
       COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL
       COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,
     2 KDTIME,TFACTR,IPER,PERIOD
       COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)
     1,JSAV1,JSAV2,ICALL,NPX,NCON
       COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),
     1    PSEDCO(10)
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),
     2 SEDFL(10,10),WATINL(10)
       COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),
     1 WAVEL(39),TOTETA(39),WLAML(39)
      REAL INTFL,NPSFL,NPSCOL
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL
      INTEGER L,E,H,B
      LOGICAL PRINT
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C EQUIVALENCE TO EXPEDITE DATA ENTRY:
      DIMENSION ABS1(39),ABS2(39),ABS3(39),ABS4(39),ABS5(39)
      EQUIVALENCE(ABS1(1),ABSG(1,1)),(ABS2(1),ABSG(1,2)),
     1 (ABS3(1),ABSG(1,3)),(ABS4(1),ABSG(1,4)),(ABS5(1),ABSG(1,5))
C-----------------------------------------------------------------------
C ASSIGN LOGICAL UNIT NUMBER FOR DATA INPUT:
      INTEGER SYSIN,REFIN
      SYSIN=5
      REFIN=7
C ASSIGN LOGICAL UNIT NUMBER FOR ERROR MESSAGE TO TERMINAL:
      TTYOUT=8
C-----------------------------------------------------------------------
```

```
C PRE-ZERO COMMONS:
C 'NPX' IS THE MAXIMUM NUMBER OF COMPARTMENTS AVAILABLE.  VALUE LOADED IADL01150
C BLOCK DATA.                                                           ADL01160
        DO 10 I=1,NPX                                                   ADL01170
        STRLDG(I)=0.                                                    ADL01180
        NPSLDG(I)=0.                                                    ADL01190
        PCPLDG(I)=0.                                                    ADL01200
        DRFLDG(I)=0.                                                    ADL01210
        IFLLDG(I)=0.                                                    ADL01220
10      CONTINUE                                                        ADL01230
C CHEMICAL DATA:                                                        ADL01240
        MWTG=0.                                                         ADL01250
        KOCG=0.                                                         ADL01260
        KOWG=0.                                                         ADL01270
        KVOG=0.                                                         ADL01280
        HENRYG=0.                                                       ADL01290
        VAPRG=0.                                                        ADL01300
        EVPRG=0.                                                        ADL01310
        DO 20 I=1,2                                                     ADL01320
        PKBS(I)=0.                                                      ADL01330
        EPKBG(I)=0.                                                     ADL01340
        PKAG(I)=0.                                                      ADL01350
        EPKAG(I)=0.                                                     ADL01360
        KCECG(I)=0.                                                     ADL01370
        KAECG(I)=0.                                                     ADL01380
20      CONTINUE                                                        ADL01390
        DO 30 I=1,5                                                     ADL01400
        SPFLG(I)=0                                                      ADL01410
        SOLG(I)=0.                                                      ADL01420
        ESOLG(I)=0.                                                     ADL01430
        KPSG(I)=0.                                                      ADL01440
        KPBG(I)=C.                                                      ADL01450
        KDPG(I)=0.                                                      ADL01460
        RFLATG(I)=40.                                                   ADL01470
30      CONTINUE                                                        ADL01480
        DO 50 I=1,5                                                     ADL01490
                DO 40 J=1,3                                             ADL01500
                QUANTG(J,I)=0.                                          ADL01510
                KAHG(J,I)=0.                                            ADL01520
                EAHG(J,I)=0.                                            ADL01530
                KNHG(J,I)=0.                                            ADL01540
                ENHG(J,I)=0.                                            ADL01550
                KBHG(J,I)=0.                                            ADL01560
                EBHG(J,I)=0.                                            ADL01570
                KOXG(J,I)=0.                                            ADL01580
                EOXG(J,I)=0.                                            ADL01590
                KBACWG(J,I)=0.                                          ADL01600
                QTBAWG(J,I)=0.                                          ADL01610
                KBACSG(J,I)=0.                                          ADL01620
                QTBASG(J,I)=0.                                          ADL01630
40              CONTINUE                                                ADL01640
50      CONTINUE                                                        ADL01650
C                                                                       ADL01660
C PRE-ZERO ENVIRONMENTAL VARIABLES:                                     ADL01670
        KOUNT=0                                                         ADL01680
        RAING=0.0                                                       ADL01690
        CLOUDG=0.                                                       ADL01700
                                                                        ADL01710
```

```
      LATG=40.                                                          ADL01720
C 'NCON' IS THE NUMBER OF INTER-COMPARTMENTAL TRANSPORT PATHWAYS THAT   ADL01730
C CAN BE SPECIFIED.  VALUE LOADED IN BLOCK DATA.                        ADL01740
      DO 60 I=1,NCON                                                    ADL01750
      JFRADG(I)=0                                                       ADL01760
      ITOADG(I)=0                                                       ADL01770
      ADVPRG(I)=0.                                                      ADL01780
      JTURBG(I)=0                                                       ADL01790
      ITURBG(I)=0                                                       ADL01800
      XSTURG(I)=0.                                                      ADL01810
      CHARLG(I)=0.                                                      ADL01820
      DSPG(I)=0.                                                        ADL01830
60    CONTINUE                                                          ADL01840
C NPX COMPARTMENTS CAN BE USED:                                         ADL01850
      DO 70 I=1,NPX                                                     ADL01860
      EVAPG(I)=0.                                                       ADL01870
      WINDG(I)=0.                                                       ADL01880
      SDCHRG(I)=0.                                                      ADL01890
      PCTWAG(I)=0.                                                      ADL01900
      FROCG(I)=0.                                                       ADL01910
      CECG(I)=0.                                                        ADL01920
      AECG(I)=0.                                                        ADL01930
      TCELG(I)=0.                                                       ADL01940
      PHG(I)=0.                                                         ADL01950
      POHG(I)=0.                                                        ADL01960
      OXRADG(I)=0.                                                      ADL01970
      BIOMSG(I)=0.                                                      ADL01980
      PLRAG(I)=0.                                                       ADL01990
      BIOTMG(I)=0.                                                      ADL02000
      BACTOG(I)=0.                                                      ADL02010
      ACBACG(I)=0.                                                      ADL02020
      KO2G(I)=0.                                                        ADL02030
      CMPETG(I)=0.                                                      ADL02040
      DOCG(I)=0.                                                        ADL02050
      CHLG(I)=0.                                                        ADL02060
      DFACG(I)=0.                                                       ADL02070
      VOLG(I)=0.                                                        ADL02080
      AREAG(I)=0.                                                       ADL02090
      DEPTHG(I)=0.                                                      ADL02100
      STFLOG(I)=0.                                                      ADL02110
      STSEDG(I)=0.                                                      ADL02120
      NPSFLG(I)=0.                                                      ADL02130
      NPSEDG(I)=0.                                                      ADL02140
      INTFLG(I)=0.                                                      ADL02150
70    CONTINUE                                                          ADL02160
      DO 80 I=1,39                                                      ADL02170
      WLAMG(I)=0.                                                       ADL02180
      DO 75 J=1,5                                                       ADL02190
C ABSORPTION OF TOXICANT:                                               ADL02200
       ABSG(I,J)=0.                                                     ADL02210
75    CONTINUE                                                          ADL02220
80    CONTINUE                                                          ADL02230
C GET NAME OF CHEMICAL, ECOSYSTEM, LOADS FROM RUN FILE:                 ADL02240
C     OPEN(UNIT=SYSIN,NAME='RUNTOX.DAT',TYPE='OLD',READONLY)            ADL02250
C**********                                                             ADL02260
C              NPRINT= NO. ITERATIONS PER PRINTOUT                      ADL02270
C              NPS, DOT = LOGICAL FLAGS FOR INPUT FROM NPS AND DIDOT    ADL02280
```

```
      READ (SYSIN,98) NPRINT,NPS,DOT                                    ADL02290
   98 FORMAT (I2,2L1)                                                   ADL02300
      WRITE (RPTOUT,99) NPS,DOT,NPRINT                                  ADL02310
   99 FORMAT(' ',79('*'),///,10X,'NON-POINT SOURCE DATA (NPS) = ',L1,//,ADL02320
     $        10X,'ATMOSPHERIC SOURCE DATA (DIDOT) = ',L1,//,           ADL02330
     $        10X,'NUMBER OF PRINTS PER EXAMS ITERATION = ',I2,///,     ADL02340
     $        ' ',79('*'))                                              ADL02350
C**********                                                             ADL02360
         READ(SYSIN,100) CHEM1                                          ADL02370
C 'CHEM1' IS 3 LETTER ABBREVIATION CODE FOR CHEMICAL TO BE RUN.         ADL02380
100      FORMAT(A3)                                                     ADL02390
C GET CODE FOR ECOSYSTEM ('ECO1'):                                      ADL02400
         READ(SYSIN,100) ECO1                                           ADL02410
C GET TIME PERIOD FOR RUN('PERIOD'):                                    ADL02420
         READ(SYSIN,103) PERIOD                                         ADL02430
103      FORMAT(F4.0)                                                   ADL02440
C**********                                                             ADL02450
       NPSPER=INT(PERIOD/24.)                                           ADL02460
       EXPERO=AMOD(PERIOD,24.)                                          ADL02470
       IF (EXPERO.NE.0.) WRITE (TTYOUT,8888)                            ADL02480
       IF (NPSPER.EQ.0) WRITE (TTYOUT,8888)                             ADL02490
       IF ((NPSPER.EQ.0).OR.(EXPERO.NE.0.)) STOP                        ADL02500
 8888  FORMAT (' NPS INPUT REQUIRES EXAMS PERIOD OF 24 HOUR MULTIPLES') ADL02510
C**********                                                             ADL02520
C-----------------------------------------------------------------------ADL02530
C FIND APPROPRIATE SECTION OF CHEMICAL DATA BASE:                       ADL02540
C        OPEN(UNIT=REFIN,NAME='TOXCHEM.DAT',TYPE='OLD',READONLY)        ADL02550
C FIND THE CHEMICAL:                                                    ADL02560
110      READ(REFIN,100,END=1000) CHEM2                                 ADL02570
         IF(CHEM2.NE.CHEM1) GO TO 110                                   ADL02580
C PROPER SECTION OF DATA BASE FOUND:                                    ADL02590
C READ NAME OF CHEMICAL:                                                ADL02600
         READ(REFIN,120) CHEMNA                                         ADL02610
C FORMAT FOR NAME OF ECOSYSTEM OR CHEMICAL:                             ADL02620
120      FORMAT(60A1)                                                   ADL02630
C                                                                       ADL02640
C READ IN THE DATA AS NEEDED:                                           ADL02650
C FLAGS TO INDICATE WHICH SPECIES OCCUR:                                ADL02660
         READ(REFIN,102) SPFLG                                          ADL02670
101   FORMAT(8F10.0)                                                    ADL02680
102   FORMAT(5(I1,1X))                                                  ADL02690
C INPUT IS SET UP IN A SERIES OF 5 BLOCKS, PROGRAM SKIPS SPECIES        ADL02700
C THAT DO NOT OCCUR:                                                    ADL02710
         IF (SPFLG(1).NE.1) GO TO 130                                   ADL02720
C-----------------------------------------------------------------------ADL02730
C READ IN DATA FOR NEUTRAL MOLECULE (SH2):                              ADL02740
       READ(REFIN,101) MWTG,SOLG(1),ESOLG(1)                            ADL02750
       READ(REFIN,101) KPSG(1),KPBG(1),KOCG,KOWG                        ADL02760
       READ(REFIN,101) KVOG,HENRYG,VAPRG,EVPRG                          ADL02770
       READ(REFIN,101) QUANTG(1,1),QUANTG(2,1),QUANTG(3,1)              ADL02780
       READ(REFIN,101) ABS1                                             ADL02790
       READ(REFIN,101) KDPG(1),RFLATG(1)                                ADL02800
       READ(REFIN,101) KAHG(1,1),KAHG(2,1),KAHG(3,1)                    ADL02810
       READ(REFIN,101) EAHG(1,1),EAHG(2,1),EAHG(3,1)                    ADL02820
       READ(REFIN,101) KNHG(1,1),KNHG(2,1),KNHG(3,1)                    ADL02830
       READ(REFIN,101) ENHG(1,1),ENHG(2,1),ENHG(3,1)                    ADL02840
       READ(REFIN,101) KBHG(1,1),KBHG(2,1),KBHG(3,1)                    ADL02850
      READ(REFIN,101) EBHG(1,1),EBHG(2,1),EBHG(3,1)
      READ(REFIN,101) KOXG(1,1),KOXG(2,1),KOXG(3,1)
      READ(REFIN,101) EOXG(1,1),EOXG(2,1),EOXG(3,1)
      READ(REFIN,101) KBACWG(1,1),KBACWG(2,1),KBACWG(3,1)
      READ(REFIN,101) QTBAWG(1,1),QTBAWG(2,1),QTBAWG(3,1)
      READ(REFIN,101) KBACSG(1,1),KBACSG(2,1),KBACSG(3,1)
      READ(REFIN,101) QTBASG(1,1),QTBASG(2,1),QTBASG(3,1)
C END OF DATA FOR SH2
130   CONTINUE
C IF NO SH3+ OCCURS, SKIP TO SH4++
      IF(SPFLG(2).NE.1) GO TO 140
C-----------------------------------------------------------------------
C DATA FOR SINGLY CHARGED CATION (SH3+)
      READ(REFIN,101) PKBG(1),EPKBG(1),SOLG(2),ESOLG(2)
      READ(REFIN,101) XPSG(2),KPBG(2),KCECG(1)
      READ(REFIN,101) QUANTG(1,2),QUANTG(2,2),QUANTG(3,2)
      READ(REFIN,101) ABS2
      READ(REFIN,101) KDPG(2),RFLATG(2)
      READ(REFIN,101) KAHG(1,2),KAHG(2,2),KAHG(3,2)
      READ(REFIN,101) EAHG(1,2),EAHG(2,2),EAHG(3,2)
      READ(REFIN,101) KNHG(1,2),KNHG(2,2),KNHG(3,2)
      READ(REFIN,101) ENHG(1,2),ENHG(2,2),ENHG(3,2)
      READ(REFIN,101) KBHG(1,2),KBHG(2,2),KBHG(3,2)
      READ(REFIN,101) EBHG(1,2),EBHG(2,2),EBHG(3,2)
      READ(REFIN,101) KOXG(1,2),KOXG(2,2),KOXG(3,2)
      READ(REFIN,101) EOXG(1,2),EOXG(2,2),EOXG(3,2)
      READ(REFIN,101) KBACWG(1,2),KBACWG(2,2),KBACWG(3,2)
      READ(REFIN,101) QTBAWG(1,2),QTBAWG(2,2),QTBAWG(3,2)
      READ(REFIN,101) KBACSG(1,2),KBACSG(2,2),KBACSG(3,2)
      READ(REFIN,101) QTBASG(1,2),QTBASG(2,2),QTBASG(3,2)
C END OF DATA ENTRY FOR SH3+
140   CONTINUE
C IF NO SH4++ OCCURS, SKIP TO SH-
      IF(SPFLG(3).NE.1) GO TO 150
C-----------------------------------------------------------------------
C DATA ENTRY FOR SH4++
      READ(REFIN,101) PKBG(2),EPKBG(2),SOLG(3),ESOLG(3)
      READ(REFIN,101) KPSG(3),KPBG(3),KCECG(2)
      READ(REFIN,101) QUANTG(1,3),QUANTG(2,3),QUANTG(3,3)
      READ(REFIN,101) ABS3
      READ(REFIN,101) KDPG(3),RFLATG(3)
      READ(REFIN,101) KAHG(1,3),KAHG(2,3),KAHG(3,3)
      READ(REFIN,101) EAHG(1,3),EAHG(2,3),EAHG(3,3)
      READ(REFIN,101) KNHG(1,3),KNHG(2,3),KNHG(3,3)
      READ(REFIN,101) ENHG(1,3),ENHG(2,3),ENHG(3,3)
      READ(REFIN,101) KBHG(1,3),KBHG(2,3),KBHG(3,3)
      READ(REFIN,101) EBHG(1,3),EBHG(2,3),EBHG(3,3)
      READ(REFIN,101) KOXG(1,3),KOXG(2,3),KOXG(3,3)
      READ(REFIN,101) EOXG(1,3),EOXG(2,3),EOXG(3,3)
      READ(REFIN,101) KBACWG(1,3),KBACWG(2,3),KBACWG(3,3)
      READ(REFIN,101) QTBAWG(1,3),QTBAWG(2,3),QTBAWG(3,3)
      READ(REFIN,101) KBACSG(1,3),KBACSG(2,3),KBACSG(3,3)
      READ(REFIN,101) QTBASG(1,3),QTBASG(2,3),QTBASG(3,3)
150   CONTINUE
C IF NO SH- OCCURS SKIP TO S=
      IF(SPFLG(4).NE.1) GO TO 160
C-----------------------------------------------------------------------
```

```
C DATA ENTRY FOR SH-
      READ(REFIN,101) PKAG(1),EPKAG(1),SOLG(4),ESOLG(4)
      READ(REFIN,101) KPSG(4),KPBG(4),KAECG(1)
      READ(REFIN,101) QUANTG(1,4),QUANTG(2,4),QUANTG(3,4)
      READ(REFIN,101) ABS4
      READ(REFIN,101) KDPG(4),RFLATG(4)
      READ(REFIN,101) KAHG(1,4),KAHG(2,4),KAHG(3,4)
      READ(REFIN,101) EAHG(1,4),EAHG(2,4),EAHG(3,4)
      READ(REFIN,101) KNHG(1,4),KNHG(2,4),KNHG(3,4)
      READ(REFIN,101) ENHG(1,4),ENHG(2,4),ENHG(3,4)
      READ(REFIN,101) KBHG(1,4),KBHG(2,4),KBHG(3,4)
      READ(REFIN,101) EBHG(1,4),EBHG(2,4),EBHG(3,4)
      READ(REFIN,101) KOXG(1,4),KOXG(2,4),KOXG(3,4)
      READ(REFIN,101) EOXG(1,4),EOXG(2,4),EOXG(3,4)
      READ(REFIN,101) KBACWG(1,4),KBACWG(2,4),KBACWG(3,4)
      READ(REFIN,101) QTBAWG(1,4),QTBAWG(2,4),QTBAWG(3,4)
      READ(REFIN,101) KBACSG(1,4),KBACSG(2,4),KBACSG(3,4)
      READ(REFIN,101) QTBASG(1,4),QTBASG(2,4),QTBASG(3,4)
C END OF DATA ENTRY FOR SH-
160   CONTINUE
C IF S= SPECIES DOES NOT OCCUR, SKIP
      IF(SPFLG(5).NE.1) GO TO 170
C-----------------------------------------------------------------------
C DATA ENTRY FOR DOUBLY CHARGED ANION S=
      READ(REFIN,101) PKAG(2),EPKAG(2),SOLG(5),ESOLG(5)
      READ(REFIN,101) KPSG(5),KPBG(5),KAECG(2)
      READ(REFIN,101) QUANTG(1,5),QUANTG(2,5),QUANTG(3,5)
      READ(REFIN,101) ABS5
      READ(REFIN,101) KDPG(5),RFLATG(5)
      READ(REFIN,101) KAHG(1,5),KAHG(2,5),KAHG(3,5)
      READ(REFIN,101) EAHG(1,5),EAHG(2,5),EAHG(3,5)
      READ(REFIN,101) KNHG(1,5),KNHG(2,5),KNHG(3,5)
      READ(REFIN,101) ENHG(1,5),ENHG(2,5),ENHG(3,5)
      READ(REFIN,101) KBHG(1,5),KBHG(2,5),KBHG(3,5)
      READ(REFIN,101) EBHG(1,5),EBHG(2,5),EBHG(3,5)
      READ(REFIN,101) KOXG(1,5),KOXG(2,5),KOXG(3,5)
      READ(REFIN,101) EOXG(1,5),EOXG(2,5),EOXG(3,5)
      READ(REFIN,101) KBACWG(1,5),KBACWG(2,5),KBACWG(3,5)
      READ(REFIN,101) QTBAWG(1,5),QTBAWG(2,5),QTBAWG(3,5)
      READ(REFIN,101) KBACSG(1,5),KBACSG(2,5),KBACSG(3,5)
      READ(REFIN,101) QTBASG(1,5),QTBASG(2,5),QTBASG(3,5)
170   CONTINUE
C END OF CHEMICAL DATA ENTRY; CLOSE THE CHEMICAL DATA BASE:
C     CLOSE(UNIT=REFIN)
C-----------------------------------------------------------------------
C ACQUIRE THE ENVIRONMENTAL DATA:
C     OPEN(UNIT=REFIN,NAME='CANON.ENV',TYPE='OLD',READONLY)
C FIND THE ECOSYSTEM:
200   READ(REFIN,100,END=1200) ECO2
      IF(ECO2.NE.ECO1) GO TO 200
C ECOSYSTEM DATA FOUND:
C READ NAME OF ECOSYSTEM:
      READ(REFIN,120) SYSTYP
C READ ENVIRONMENTAL DATA FROM THE FILE:
      READ(REFIN,201) KOUNT
201   FORMAT(16I5)
      READ(REFIN,210) (TYPEE(J),J=1,KOUNT)
```

```
  210   FORMAT(A1)                                                      ADL04000
        READ(REFIN,101) LATG                                            ADL04010
        READ(REFIN,101) (WINDG(J),J=1,KOUNT)                            ADL04020
        READ(REFIN,101) (VOLG(J),J=1,KOUNT)                             ADL04030
        READ(REFIN,101) (AREAG(J),J=1,KOUNT)                            ADL04040
        READ(REFIN,101) (DEPTHG(J),J=1,KOUNT)                           ADL04050
        READ(REFIN,101) (STFLOG(J),J=1,KOUNT)                           ADL04060
        READ(REFIN,101) (STSEDG(J),J=1,KOUNT)                           ADL04070
        READ(REFIN,101) (NPSFLG(J),J=1,KOUNT)                           ADL04080
        READ(REFIN,101) (NPSEDG(J),J=1,KOUNT)                           ADL04090
        READ(REFIN,101) (INTFLG(J),J=1,KOUNT)                           ADL04100
        READ(REFIN,101) RAING                                           ADL04110
        READ(REFIN,101) CLOUDG                                          ADL04120
        READ(REFIN,101) WLAMG                                           ADL04130
        READ(REFIN,101) (DFACG(J),J=1,KOUNT)                            ADL04140
        READ(REFIN,101) (EVAPG(J),J=1,KOUNT)                            ADL04150
        READ(REFIN,101) (SDCHRG(J),J=1,KOUNT)                           ADL04160
        READ(REFIN,101) (PCTWAG(J),J=1,KOUNT)                           ADL04170
        READ(REFIN,201) JFRADG                                          ADL04180
        READ(REFIN,201) ITOADG                                          ADL04190
        READ(REFIN,101) ADVPRG                                          ADL04200
        READ(REFIN,201) JTURBG                                          ADL04210
        READ(REFIN,201) ITURBG                                          ADL04220
        READ(REFIN,101) XSTURG                                          ADL04230
        READ(REFIN,101) CHARLG                                          ADL04240
        READ(REFIN,101) DSPG                                            ADL04250
        READ(REFIN,101) (FROCG(J),J=1,KOUNT)                            ADL04260
        READ(REFIN,101) (CECG(J),J=1,KOUNT)                             ADL04270
        READ(REFIN,101) (AECG(J),J=1,KOUNT)                             ADL04280
        READ(REFIN,101) (TCELG(J),J=1,KOUNT)                            ADL04290
        READ(REFIN,101) (PHG(J),J=1,KOUNT)                              ADL04300
        READ(REFIN,101) (POHG(J),J=1,KOUNT)                             ADL04310
        READ(REFIN,101) (OXRADG(J),J=1,KOUNT)                           ADL04320
        READ(REFIN,101) (BIOMSG(J),J=1,KOUNT)                           ADL04330
        READ(REFIN,101) (PLRAG(J),J=1,KOUNT)                            ADL04340
        READ(REFIN,101) (BIOTMG(J),J=1,KOUNT)                           ADL04350
        READ(REFIN,101) (BACTOG(J),J=1,KOUNT)                           ADL04360
        READ(REFIN,101) (ACBACG(J),J=1,KOUNT)                           ADL04370
        READ(REFIN,101) (KO2G(J),J=1,KOUNT)                             ADL04380
        READ(REFIN,101) (CMPETG(J),J=1,KOUNT)                           ADL04390
        READ(REFIN,101) (DOCG(J),J=1,KOUNT)                             ADL04400
        READ(REFIN,101) (CHLG(J),J=1,KOUNT)                             ADL04410
C END OF ENVIRONMENTAL DATA ENTRY                                        ADL04420
C       CLOSE(UNIT=REFIN)                                               ADL04430
C-----------------------------------------------------------------------ADL04440
        CALL GHOST                                                      ADL04450
C                                                                       ADL04460
C**********                                                             ADL04470
C                                                                       ADL04480
C                                                                       ADL04490
C     THE FOLLOWING RETURN STATEMENT IS NOT VALID ON THE PDP-11         ADL04500
C                                                                       ADL04510
C      RETURN                                                           ADL04520
C                                                                       ADL04530
C**********                                                             ADL04540
        STOP                                                            ADL04550
C-----------------------------------------------------------------------ADL04560
```

```
C DATA ON REQUESTED CHEMICAL NOT AVAILABLE:
C CLOSE THE CHEMICAL DATA FILE:
C1000     CLOSE(UNIT=REFIN)
1000     CONTINUE
        WRITE(TTYOUT,1100) CHEM1
1100     FORMAT(1X,'ERROR: NO DATA FOUND FOR CHEMICAL ',A3)
        STOP
C ECOSYSTEM NOT FOUND IN CANONICAL ENVIRONMENTS FILE:
C1200    CLOSE(UNIT=REFIN)
1200  CONTINUE
      WRITE(TTYOUT,1300) ECO1
1300  FORMAT(1X,'ERROR: NO DATA FOUND FOR ECOSYSTEM ',A3)
      STOP
      END
C FILE 'GHOST.FTN'
C  EXAMS-AETOX1: A STEADY-STATE MODEL OF TRANSPORT AND TRANSFORMATIONS
C  OF TOXIC ORGANIC CHEMICALS IN AQUATIC ECOSYSTEMS.
C AUTHOR:        LAWRENCE A. BURNS
C                ATHENS ENVIRONMENTAL RESEARCH LABORATORY
C                ENVIRONMENTAL SYSTEMS BRANCH
C                U.S. ENVIRONMENTAL PROTECTION AGENCY
C                COLLEGE STATION ROAD
C                ATHENS, GEORGIA 30605
C                U.S.A.
C                TEL: (404)546-3148      (FTS) 250-3148
C-----------------------------------------------------------------
      SUBROUTINE GHOST
C GHOST. GHOST IS PRIMARILY A PROGRAM CONTROL MODULE.  IT PERFORMS FEW
C COMPUTATIONS ASIDE FROM PRESERVING INTERMEDIATE VARIABLES AND
C INITIALIZING RUN-CONTROL PARAMETERS.  GHOST OPENS THE FILES FOR
C PROGRAM OUTPUT, WRITES INFORMATION TO THE FILES FOR IDENTIFYING THE
C RUN, CALLS SUBROUTINES, AND CHECKS FOR ERROR FLAGS THAT MAY BE SET
C BY THE SUBROUTINES.
C
C-----------------------------------------------------------------------
C SUBROUTINES USED BY INTEGRATION ROUTINE:
      EXTERNAL FCT,OUTP
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG
     1,NPSEDG,INTFLG
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT
      LOGICAL*1 CHEMNA,SYSTYP
      COMMON/NAME1G/CHEMNA(60)
      COMMON/CONTRG/SPFLG(5)
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)
     1,KBHG(3,5),EBHG(3,5)
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)
     1,IFLLDG(10)
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),B!OMSG(10)
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG
     1,DFACG(10)
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)
     1,INTFLG(10)
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:
C-----------------------------------------------------------------------
C FILE 'CHEML.COM'
      COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)
      COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)
      COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)
      COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV
      COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),
     2  CHEMPC,BIOPCT,VOLPCT,EXPPCT
      COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L
C-----------------------------------------------------------------------
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:
C-----------------------------------------------------------------------
C FILE 'ENVIRL.COM'
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),
     2  NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,
     2  KDTIME,TFACTR,IPER,PERIOD
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)
     1,JSAV1,JSAV2,ICALL,NPX,NCON
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),
     1     PSEDCO(10)
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),
     2 SEDFL(10,10),WATINL(10)
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),
     1  WAVEL(39),TOTETA(39),WLAML(39)
      REAL INTFL,NPSFL,NPSCOL
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL
      LOGICAL PRINT
      INTEGER L,E,H,B
C**********
      REAL NPSRO,NPSRO1
      LOGICAL NPS,DOT
C**********
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
```

```
C Y IS TOXICANT CONCENTRATION REFERRED TO AQUEOUS PHASE OF SYSTEM.
C VERSION IS DIMENSIONED FOR 10 COMPARTMENTS:
      DIMENSION Y(10)
C**********
      COMMON/NPSBLK/NPS,LUNPS,LUNPS2,NPSPER
      COMMON/DOTBLK/DOT,LUDOT
      DIMENSION NPSRO1(10),DOTRO1(10),DOTRO2(10),NPSRO(10),DOTROW(10)
     $         ,DOTROD(10),FLO1(10),SED1(10)
C**********
       INTEGER SYSIN
       SYSIN=5
C-----------------------------------------------------------------------
C INITIALIZE FLAGS FOR MONITORING CALLS TO PHOTOLYSIS SUBROUTINES:
       JSAV1=0
       JSAV2=0
       ICALL=0
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C PARAMETERS FOR INTEGRATION SUBROUTINE:
       T=0.
       ABSERR=1.E-6
       RELERR=1.E-6
C TPRINT SET TO ZERO TO PRINT INITIAL CONDITIONS:
       TPRIN1=0.
       IFLAG=1
C-----------------------------------------------------------------------
C CHECK TO ENSURE THAT THE NUMBER OF COMPARTMENTS REQUESTED BY THE
C USER DOES NOT EXCEED THE MAXIMUM SIZE OF THE MODEL:
C VERSION IS DIMENSIONED FOR 10 COMPARTMENTS:
       NP=10
       IF(KOUNT.LE.NP) GO TO 110
       WRITE(TTYOUT,100) NP,KOUNT
100    FORMAT(1X,'THE MODEL IS SET UP FOR ',I4,' COMPARTMENTS AT',
     2 ' MOST:'/1X,'YOU REQUESTED ',I4,'.  NO SIMULATION PERFORMED.')
       GO TO 600
110    CONTINUE
C-----------------------------------------------------------------------
C OPEN FILE FOR REPORT ON TOXICANT IN A SPECIFIED SITUATION:
C      OPEN(UNIT=RPTOUT,NAME='REPORT.DAT')
C PAGE HEADER: NAME OF CHEMICAL AND ECOSYSTEM
130    FORMAT(1H1,'  AERL-ESB MODEL OF FATE OF ORGANIC TOXICANTS',
     11X,'IN AQUATIC ECOSYSTEMS.'/1X,'CHEMICAL: ',60A1/1X,
     2'ECOSYSTEM: ',60A1/1X,72(1H-))
C-----------------------------------------------------------------------
C OPEN PLOTTING FILES USED IN MODEL:
C  TIME TRACE OF MODEL OUTPUT:
C      OPEN (UNIT=KINOUT,NAME='KINE.DAT',TYPE='NEW',FORM='UNFORMATTED')
C  FILE FOR PLOTTING STEADY-STATE CONCENTRATIONS:
C      OPEN(UNIT=SSOUT,NAME='EXPO.DAT',TYPE='NEW',FORM='UNFORMATTED')
C IDENTIFY TIME TRACE WRITTEN IN 'KINE.DAT':
       WRITE(KINOUT) CHEMNA,SYSTYP
C IDENTIFY INFORMATION WRITTEN TO 'EXPO.DAT':
       WRITE(SSOUT) CHEMNA,SYSTYP
C-----------------------------------------------------------------------
C DOCUMENT INPUT CHARACTERISTICS OF COMPOUND:
       CALL PRCHEM
C-----------------------------------------------------------------------
```

```
C DOCUMENT INPUT DATA DESCRIBING THE ENVIRONMENT:
C WRITE PAGE HEADER:
      IF (PRINT) WRITE(RPTOUT,130) CHEMNA,SYSTYP
      CALL PRENV
C-----------------------------------------------------------------------
C CONVERT INPUT ENVIRONMENTAL DATA TO QUANTITIES USED IN MODEL AND
C DOCUMENT FLOW CHARACTERISTICS OF ENVIRONMENT:
C WRITE PAGE HEADER:
      IF (PRINT) WRITE(RPTOUT,130) CHEMNA,SYSTYP
      CALL FLOWS
C ABORT SYSTEM IF NECCESARY:
      IF(IFLAG.EQ.8) GO TO 500
C-----------------------------------------------------------------------
C COMPUTE DISTRIBUTION OF TOXICANT AMONG CHEMICAL SPECIES & FORMS:
      CALL DISTRB
C-----------------------------------------------------------------------
C SET INITIAL PERIOD COUNT IPER
      IPER=1
C READ IN LOADING VECTORS (ALL AS KG OF TOXICANT PER HOUR):
C STREAM LOADINGS:
140     READ(SYSIN,101,END=600) STRLDG
101   FORMAT(8F10.0)
C**********
C
              PRINT=MOD(IPER-1,NPRINT).EQ.0
C
      IF (PRINT) WRITE (RPTOUT,102) IPER
  102 FORMAT ('1',///////////////////,
     $        30X,'ITERATION NUMBER = ',I4)
C
      DO 141 ICELL=1,10
      NPSRO(ICELL)=0.
      DOTROW(ICELL)=0.
  141 DOTROD(ICELL)=0.
C
      IF (.NOT.NPS) GO TO 143
C
C             ZERO NON-POINT SOURCE FLOW AND SEDIMENT LOADINGS
C             IF THEY ARE TO COME FROM NPS
      DO 146 ICELL=1,10
      NPSFLG(ICELL)=0.
  146 NPSEDG(ICELL)=0.
C
  143 IF ((.NOT.NPS).AND.(.NOT.DOT)) GO TO 147
C
C             THIS SECTION INCREMENTS LOADINGS ACCORDING TO
C             THE OUTPUT FROM NPS AND DIDOT.
C             ALL VARIABLES ENDING IN 1 ARE TEMPORARY DAILY VALUES
C             WHICH WILL BE SUMMED OVER THE NUMBER OF NPS PERIODS PER
C             EXAMS LOADING PERIODS.
C             NPSRO IS THE TOTAL RUNOFF LOADING FROM NPS.
C             DOTROD AND DOTROW ARE THE DRY AND WET LOADINGS FROM DIDOT.
      AVGING=1./(24.*NPSPER)
      DO 142 N=1,NPSPER
      IF (NPS) READ (LUNPS,145,END=600) NPSRO1
      IF (NPS) READ (LUNPS2,145,END=550) FLO1
      IF (NPS) READ (LUNPS2,145,END=550) SED1                           ADL06850
C        JDOT IS A RAIN FLAG INDICATING A WET DEPOSITION RECORD FOLLOWS ADL06860
      JDOT=0                                                            ADL06870
      IF (DOT) READ (LUDOT,144,END=550) DOTRO1,JDOT                     ADL06880
      IF (JDOT.EQ.1) READ (LUDOT,144,END=550) DOTRO2                    ADL06890
C                                                                       ADL06900
C        INCREMENT LOADINGS AND AVERAGE OVER THE PERIOD                 ADL06910
C        TO GET MEAN KG/HR.                                             ADL06920
      DO 142 ICELL=1,10                                                 ADL06930
      NPSEDG(ICELL)=NPSEDG(ICELL)+AVGING*SED1(ICELL)                    ADL06940
      NPSFLG(ICELL)=NPSFLG(ICELL)+AVGING*FLO1(ICELL)                    ADL06950
      NPSRO(ICELL)=NPSRO(ICELL)+AVGING*NPSRO1(ICELL)                    ADL06960
      DOTROD(ICELL)=DOTROD(ICELL)+AVGING*DOTRO1(ICELL)                  ADL06970
      IF (JDOT.EQ.1) DOTROW(ICELL)=DOTROW(ICELL)+AVGING*DOTRO2(ICELL)   ADL06980
  142 CONTINUE                                                          ADL06990
C                                                                       ADL07000
144   FORMAT(1X,10E11.4,I1)                                             ADL07010
145   FORMAT (10E11.4)                                                  ADL07020
  147 CONTINUE                                                          ADL07030
C                                                                       ADL07040
C                                                                       ADL07050
C**********                                                             ADL07060
C NON-POINT-SOURCE LOADINGS:                                            ADL07070
         READ(SYSIN,101,END=550) NPSLDG                                 ADL07080
C PRECIPITATION (RAINOUT) LOADINGS:                                     ADL07090
         READ(SYSIN,101,END=550) PCPLDG                                 ADL07100
C DRIFT LOADINGS:                                                       ADL07110
         READ(SYSIN,101,END=550) DRFLDG                                 ADL07120
C INTERFLOW LOADINGS:                                                   ADL07130
         READ(SYSIN,101,END=550) IFLLDG                                 ADL07140
C**********                                                             ADL07150
C OBTAIN VALUES FOR 'PSEDCO'                                            ADL07160
         READ(SYSIN,101,END=999) PSEDCO                                 ADL07170
C**********                                                             ADL07180
C  ADD ON NPS AND DIDOT LOADINGS, WEIGHTING DIDOT'S BY                  ADL07190
C  THE AREA OF EACH CELL.                                               ADL07200
 999  DO 148 ICELL=1,10                                                 ADL07210
      DRFLDG(ICELL)=DRFLDG(ICELL)+AREAG(ICELL)*DOTROD(ICELL)            ADL07220
      PCPLDG(ICELL)=PCPLDG(ICELL)+AREAG(ICELL)*DOTROW(ICELL)            ADL07230
      IF (PSEDCO (ICELL) .LE. 0.0) PSEDCO (ICELL) = 25.0                ADL07240
  148 NPSLDG(ICELL)=NPSLDG(ICELL)+NPSRO(ICELL)                          ADL07250
C**********                                                             ADL07260
C NAME OF CHEMICAL, ECOSYSTEM, LOADS ACQUIRED:                          ADL07270
C        CLOSE(UNIT=SYSIN)                                              ADL07280
C-----------------------------------------------------------------------ADL07290
C EVALUATE PLAUSIBILITY OF LOADINGS TO SYSTEM:                          ADL07300
         CALL CKLOAD                                                    ADL07310
C MODEL NEED NOT BE RUN IF ALL LOADS DISAPPEAR:                         ADL07320
         IF (IFLAG.EQ.8) GO TO 400                                      ADL07330
C-----------------------------------------------------------------------ADL07340
C COMPUTE FIRST-ORDER RATE COEFFICIENTS FOR PROCESSES:                  ADL07350
C WRITE PAGE HEADER FOR TABLE OF PSEUDO-FIRST-ORDER KINETICS:           ADL07360
         IF (PRINT.AND.(.NOT.NPS).AND.(.NOT.DOT))                       ADL07370
     $      WRITE (RPTOUT,130) CHEMNA,SYSTYP                            ADL07380
         CALL FIRORD                                                    ADL07390
C-----------------------------------------------------------------------ADL07400
C COMPUTE STEADY-STATE CONCENTRATIONS IN SYSTEM ELEMENTS:               ADL07410
```

```
C PAGE HEADER FOR LOAD PROFILE:
        IF (PRINT) WRITE(RPTOUT,130) CHEMNA,SYSTYP
        CALL STEADY(Y)
C PROBLEMS WITH DRIFT LOADS MAY BE ENCOUNTERED IN STEADY-STATE ROUTINE:
        IF (IFLAG.GE.8) GO TO 400
C-----------------------------------------------------------------------
C REPORT ON GENERAL DISTRIBUTION AND FATE OF TOXICANT:
        IF(IPER.EQ.1) CALL AVEOUT(Y)
C-----------------------------------------------------------------------
C ANALYZE AND REPORT MASS FLUX RATIOS:
150     CALL FLXOUT(Y)
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C STEADY-STATE ANALYSIS COMPLETE; TIME FRAME FOR DECAY SIMULATION
C COMPUTED FROM STEADY-STATE PROPERTIES.
C RUN DECAY LIMB OF SIMULATION:
C  SAVE STEADY-STATE POLLUTANT MASSES:
C  POLLUTANT MASS IN WATER COLUMN:
        QWSAV=Z(5)
C MASS IN SEDIMENTS:
        QSSAV=Z(6)
C TOTAL POLLUTANT MASS IN SYSTEM:
        QTSAV=Z(5)+Z(6)
C SET UP PAGE HEADER FOR REPORT ON DECAY SIMULATION:
C**********
      IF (.NOT.PRINT) GO TO 225
C**********
        WRITE(RPTOUT,130) CHEMNA,SYSTYP
        WRITE(RPTOUT,160)
160     FORMAT(15X,'SIMULATION OF SYSTEM RESPONSE AFTER LOAD CEASES'/
     2 1X,72(1H-))
C COLUMN HEADERS:
        WRITE(RPTOUT,170)
        IF(KDTIME.EQ.1) WRITE(RPTOUT,180)
        IF(KDTIME.EQ.2) WRITE(RPTOUT,190)
        IF(KDTIME.EQ.3) WRITE(RPTOUT,200)
        IF(KDTIME.EQ.4) WRITE(RPTOUT,210)
        WRITE(RPTOUT,220)
170     FORMAT(1X,'TIME',7X,'AVERAGE POLLUTANT CONCENTRATIONS',10X,
     1 'MASS OF POLLUTANT'/1X,5(1H-),6X,32(1H-),10X,17(1H-))
180     FORMAT(1X,'HOURS',4X, 'WATER COLUMN',9X,'BOTTOM SEDIMENTS',
     1 5X,'WATER COL',3X,'SEDIMENTS')
190     FORMAT(1X,'DAYS',5X, 'WATER COLUMN',9X,'BOTTOM SEDIMENTS',
     1 5X,'WATER COL',3X,'SEDIMENTS')
200     FORMAT(1X,'MONTH',4X, 'WATER COLUMN',9X,'BOTTOM SEDIMENTS',
     1 5X,'WATER COL',3X,'SEDIMENTS')
210     FORMAT(1X,'YEARS',4X, 'WATER COLUMN',9X,'BOTTOM SEDIMENTS',
     1 5X,'WATER COL',3X,'SEDIMENTS')
220     FORMAT(6X,21(1H-),2X,21(1H-),2X,9(1H-),3X,9(1H-)/6X,
     1 'FREE(MG/L) SED(MG/KG)  PORE(MG/L) SED(MG/KG)  TOTAL ',
     2 'KG    TOTAL KG'/)
C-----------------------------------------------------------------------
C**********
225     CONTINUE
C**********
C CALL INTEGRATOR:
        CALL RKF45(Y,FCT,OUTP)
```

```
C IF SIMULATION FAILED, RETURN - -                                      ADL07990
      IF(IFLAG.LT.8) GO TO 240                                          ADL08000
      WRITE(TTYOUT,230)                                                 ADL08010
230   FORMAT(1X,'INTEGRATOR FAILED - PERSISTENCE INFORMATION',          ADL08020
    2  ' NOT COMPUTED.')                                                ADL08030
C ANALYSIS OF KINETICS (IN 'SUMUP') SKIPPED IF IFLAG.GE.8               ADL08040
240   CONTINUE                                                          ADL08050
C-----------------------------------------------------------------------ADL08060
C PAGE HEADER FOR EXPOSURE SUMMARY:                                     ADL08070
      IF (PRINT) WRITE(RPTOUT,130) CHEMNA,SYSTYP                        ADL08080
C CALL ROUTINE TO ANALYZE KINETICS AND SUMMARIZE RESULTS:               ADL08090
      IF (QTSAV.GT.0.) CALL SUMUP                                       ADL08100
      IPER = IPER + 1                                                   ADL08110
      PERIOD =(PERIOD * IPER)/(IPER-1)                                  ADL08120
      IFLAG = 1                                                         ADL08130
    GO TO 140                                                           ADL08140
C-----------------------------------------------------------------------ADL08150
C REPORT COMPLETE FOR THIS ECOSYSTEM:                                   ADL08160
C400     CLOSE(UNIT=RPTOUT,DISPOSE='SAVE')                              ADL08170
400   CONTINUE                                                          ADL08180
C        CLOSE(UNIT=KINOUT,DISPOSE='SAVE')                              ADL08190
C        CLOSE(UNIT=SSOUT,DISPOSE='SAVE')                               ADL08200
C-----------------------------------------------------------------------ADL08210
      GO TO 600                                                         ADL08220
C-----------------------------------------------------------------------ADL08230
C SYSTEM ABORTED:                                                       ADL08240
C500     CLOSE(UNIT=RPTOUT,DISPOSE='DELETE')                            ADL08250
500   CONTINUE                                                          ADL08260
C        CLOSE(UNIT=KINOUT,DISPOSE='DELETE')                            ADL08270
C        CLOSE(UNIT=SSOUT,DISPOSE='DELETE')                             ADL08280
         GO TO 600                                                      ADL08290
C-----------------------------------------------------------------------ADL08300
550      CONTINUE                                                       ADL08310
         WRITE(TTYOUT,551)                                              ADL08320
551      FORMAT(1X,'ERROR: INCOMPLETE LOADING INPUT')                   ADL08330
600      CONTINUE                                                       ADL08340
         RETURN                                                         ADL08350
         END                                                            ADL08360
         SUBROUTINE INREC                                               ADL08370
         RETURN                                                         ADL08380
         END                                                            ADL08390
      SUBROUTINE PRCHEM                                                 ADL00010
C-----------------------------------------------------------------------ADL00020
C PRCHEM. PRCHEM RECORDS THE CHEMICAL CHARACTERIZATION OF THE TOXICANT  ADL00030
C (INPUT DATA) IN THE REPORT FILE ('REPORT.DAT').                       ADL00040
C-----------------------------------------------------------------------ADL00050
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL00060
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL00070
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL00080
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG     ADL00090
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL00100
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL00110
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL00120
     1,NPSEDG,INTFLG                                                    ADL00130
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH     ADL00140
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL00150
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL00160
      COMMON/NAME1G/CHEMNA(60)                                          ADL00170
      COMMON/CONTRG/SPFLG(5)                                            ADL00180
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL00190
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL00200
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL00210
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL00220
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL00230
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL00240
     1,KBHG(3,5),EBHG(3,5)                                              ADL00250
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL00260
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL00270
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL00280
     1,IFLLDG(10)                                                       ADL00290
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL00300
      COMMON/SET PG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL00310
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL00320
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL00330
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)     ADL00340
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)              ADL00350
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG         ADL00360
     1,DFACG(10)                                                        ADL00370
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                        ADL00380
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                       ADL00390
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)          ADL00400
     1,INTFLG(10)                                                       ADL00410
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL00420
C-----------------------------------------------------------------------ADL00430
C FILE 'CHEML.COM'                                                      ADL00440
      COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)            ADL00450
      COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),        ADL00460
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                    ADL00470
      COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                         ADL00480
      COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                             ADL00490
      COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),             ADL00500
     2  CHEMPC,BIOPCT,VOLPCT,EXPPCT                                     ADL00510
      COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)                 ADL00520
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                         ADL00530
C-----------------------------------------------------------------------ADL00540
C-----------------------------------------------------------------------ADL00550
C FILE 'ENVIRL.COM'                                                     ADL00560
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                    ADL00570
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),
     2  NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,
     2  KDTIME,TFACTR,IPER,PERIOD
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)
     1,JSAV1,JSAV2,ICALL,NPX,NCON
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),
     1     PSEDCO(10)
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),
     2 SEDFL(10,10),WATINL(10)
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),
     1  WAVEL(39),TOTETA(39),WLAML(39)
      REAL INTFL,NPSFL,NPSCOL
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL
      INTEGER L,E,H,B
      LOGICAL PRINT
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C ONE PAGE IS PRINTED FOR EACH CHEMICAL SPECIES (SH2, ETC.)
C-----------------------------------------------------------------------
C**********
      IF (.NOT.PRINT) GO TO 600
C**********
      IF(SPFLG(1).NE.1) GO TO 200
C-----------------------------------------------------------------------
C PRINT DATA FOR NEUTRAL MOLECULE (SH2):
C -------------------------------------
C PAGE HEADER: NAME OF CHEMICAL AND ECOSYSTEM
      WRITE(RPTOUT,100) CHEMNA,SYSTYP
100   FORMAT(1H1,'  AERL-ESB MODEL OF FATE OF ORGANIC TOXICANTS',
     11X,'IN AQUATIC ECOSYSTEMS.'/1X,'CHEMICAL: ',60A1/1X,
     2'ECOSYSTEM: ',60A1/1X,72(1H-))
      WRITE(RPTOUT,110)
110   FORMAT(1X,'SH2 (NEUTRAL MOLECULE, SPECIES #1) INPUT DATA:')
      WRITE(RPTOUT,115) MWTG,SOLG(1),ESOLG(1),RFLATG(1)
115   FORMAT(1X,'MWT=',1PG11.4,' SOL=',G11.4,' ESOL=',G11.4
     1,' RFLAT=',G11.4)
      WRITE(RPTOUT,120) KVOG,HENRYG,VAPRG,EVPRG
120   FORMAT(1X,'KVO=',1PG11.4,' HENRY=',G11.4,' VAPR=',G11.4,
     2  ' EVPR=',G11.4)
      WRITE(RPTOUT,125) KPSG(1),KPBG(1),KOCG,KOWG
125   FORMAT(1X,'KPS=',1PG11.4,' KPB=',G11.4,' KOC=',G11.4,' KOW=',
     2  G11.4)
      WRITE(RPTOUT,130) KDPG(1),QUANTG(1,1),QUANTG(2,1),QUANTG(3,1)
130   FORMAT(1X,'KDP=',1PG11.4,' QUANT1=',G11.4,' QUANT2=',G11.4,
     2  ' QUANT3=',G11.4)
      WRITE(RPTOUT,135) KAHG(1,1),EAHG(1,1),KNHG(1,1),ENHG(1,1)
135   FORMAT(1X,'KAH1=',1PG11.4,' EAH1=',G11.4,'  KNH1=',G11.4,
     2  ' ENH1=',G11.4)
      WRITE(RPTOUT,140) KAHG(2,1),EAHG(2,1),KNHG(2,1),ENHG(2,1)
140   FORMAT(1X,'KAH2=',1PG11.4,' EAH2=',G11.4,'  KNH2=',G11.4,
     2  ' ENH2=',G11.4)
      WRITE(RPTOUT,145) KAHG(3,1),EAHG(3,1),KNHG(3,1),ENHG(3,1)
145   FORMAT(1X,'KAH3=',1PG11.4,' EAH3=',G11.4,'  KNH3=',G11.4,
     2  ' ENH3=',G11.4)
      WRITE(RPTOUT,150) KBHG(1,1),EBHG(1,1),KOXG(1,1),EOXG(1,1)
150   FORMAT(1X,'KBH1=',1PG11.4,' EBH1=',G11.4,'  KOX1=',G11.4,
     2  ' EOX1=',G11.4)
        WRITE(RPTOUT,155) KBHG(2,1),EBHG(2,1),KOXG(2,1),EOXG(2,1)
155     FORMAT(1X,'KBH2=',1PG11.4,' EBH2=',G11.4,'  KOX2=',G11.4,
     2  ' EOX2=',G11.4)
        WRITE(RPTOUT,160) KBHG(3,1),EBHG(3,1),KOXG(3,1),EOXG(3,1)
160     FORMAT(1X,'KBH3=',1PG11.4,' EBH3=',G11.4,'  KOX3=',G11.4,
     2  ' EOX3=',G11.4)
        WRITE(RPTOUT,175)KBACWG(1,1),QTBAWG(1,1),KBACSG(1,1),QTBASG(1,1)
175     FORMAT(1X,'KBACW1=',1PG11.4,' QTW1=',G11.4,2X,'KBACS1=',
     2  G11.4,' QTS1=',G11.4)
        WRITE(RPTOUT,180)KBACWG(2,1),QTBAWG(2,1),KBACSG(2,1),QTBASG(2,1)
180     FORMAT(1X,'KBACW2=',1PG11.4,' QTW2=',G11.4,2X,'KBACS2=',
     2  G11.4,' QTS2=',G11.4)
        WRITE(RPTOUT,185)KBACWG(3,1),QTBAWG(3,1),KBACSG(3,1),QTBASG(3,1)
185     FORMAT(1X,'KBACW3=',1PG11.4,' QTW3=',G11.4,2X,'KBACS3=',
     2  G11.4,' QTS3=',G11.4)
        WRITE(RPTOUT,190) (ABSG(I,1),I=1,3)
190     FORMAT(1X,'ABSORPTION SPECTRUM (ABS):',3(2X,1PG11.4))
        WRITE(RPTOUT,195) (ABSG(I,1),I=4,9)
        WRITE(RPTOUT,195) (ABSG(I,1),I=10,15)
        WRITE(RPTOUT,195) (ABSG(I,1),I=16,21)
        WRITE(RPTOUT,195) (ABSG(I,1),I=22,27)
        WRITE(RPTOUT,195) (ABSG(I,1),I=28,33)
        WRITE(RPTOUT,195) (ABSG(I,1),I=34,39)
195     FORMAT(6(1X,1PG11.4))
C-----------------------------------------------------------------------
200     CONTINUE
        IF(SPFLG(2).NE.1) GO TO 300
C-----------------------------------------------------------------------
C PRINT DATA FOR SH3+
C -------------------
C PAGE HEADER: NAME OF CHEMICAL AND ECOSYSTEM
        WRITE(RPTOUT,100) CHEMNA,SYSTYP
        WRITE(RPTOUT,210)
210   FORMAT(1X,'SH3+ (SINGLY CHARGED CATION, SPECIES #2) INPUT DATA:')
        WRITE(RPTOUT,215) PKBG(1),EPKBG(1),SOLG(2),ESOLG(2)
215     FORMAT(1X,'PKB1=',1PG11.4,' EPKB1=',G11.4,' SOL=',G11.4,
     2  ' ESOL=',G11.4)
        WRITE(RPTOUT,220) MWTG
220     FORMAT(1X,'MWT=',1PG11.4)
        WRITE(RPTOUT,225) KP3G(2),KPBG(2),KCECG(1),RFLATG(2)
225     FORMAT(1X,'KPS=',1PG11.4,' KPB=',G11.4,' KCEC=',G11.4
     1,' RFLAT=',G11.4)
        WRITE(RPTOUT,130) KDPG(2),QUANTG(1,2),QUANTG(2,2),QUANTG(3,2)
        WRITE(RPTOUT,135) KAHG(1,2),EAHG(1,2),KNHG(1,2),ENHG(1,2)
        WRITE(RPTOUT,140) KAHG(2,2),EAHG(2,2),KNHG(2,2),ENHG(2,2)
        WRITE(RPTOUT,145) KAHG(3,2),EAHG(3,2),KNHG(3,2),ENHG(3,2)
        WRITE(RPTOUT,150) KBHG(1,2),EBHG(1,2),KOXG(1,2),EOXG(1,2)
        WRITE(RPTOUT,155) KBHG(2,2),EBHG(2,2),KOXG(2,2),EOXG(2,2)
        WRITE(RPTOUT,160) KBHG(3,2),ERHG(3,2),KOXG(3,2),EOXG(3,2)
        WRITE(RPTOUT,175)KBACWG(1,2),QTBAWG(1,2),KBACSG(1,2),QTBASG(1,2)
        WRITE(RPTOUT,180)KBACWG(2,2),QTBAWG(2,2),KBACSG(2,2),QTBASG(2,2)
        WRITE(RPTOUT,185)KBACWG(3,2),QTBAWG(3,2),KBACSG(3,2),QTBASG(3,2)
        WRITE(RPTOUT,190) (ABSG(I,2),I=1,3)
        WRITE(RPTOUT,195) (ABSG(I,2),I=4,9)
        WRITE(RPTOUT,195) (ABSG(I,2),I=10,15)
        WRITE(RPTOUT,195) (ABSG(I,2),I=16,21)
```

```
      WRITE(RPTOUT,195) (ABSG(I,2),I=22,27)                              ADL01720
      WRITE(RPTOUT,195) (ABSG(I,2),I=28,33)                              ADL01730
      WRITE(RPTOUT,195) (ABSG(I,2),I=34,39)                              ADL01740
C-----------------------------------------------------------------------  ADL01750
300   CONTINUE                                                           ADL01760
      IF(SPFLG(3).NE.1) GO TO 400                                        ADL01770
C-----------------------------------------------------------------------  ADL01780
C PRINT DATA FOR SH4++                                                   ADL01790
C --------------------                                                   ADL01800
C PAGE HEADER: NAME OF CHEMICAL AND ECOSYSTEM                            ADL01810
      WRITE(RPTOUT,100) CHEMNA,SYSTYP                                    ADL01820
      WRITE(RPTOUT,310)                                                  ADL01830
310   FORMAT(1X,'SH4++ (DOUBLY CHARGED CATION, SPECIES #3) INPUT DATA:')ADL01840
      WRITE(RPTOUT,315) PKBG(2),EPKBG(2),SOLG(3),ESOLG(3)                ADL01850
315   FORMAT(1X,'PKB2=',1PG11.4,' EPKB2=',G11.4,' SOL=',G11.4,          ADL01860
     2 ' ESOL=',G11.4)                                                   ADL01870
      WRITE(RPTOUT,220) MWTG                                             ADL01880
      WRITE(RPTOUT,225) KPSG(3),KPBG(3),KCECG(2),RFLATG(3)               ADL01890
      WRITE(RPTOUT,130) KDPG(3),QUANTG(1,3),QUANTG(2,3),QUANTG(3,3)      ADL01900
      WRITE(RPTOUT,135) KAHG(1,3),EAHG(1,3),KNHG(1,3),ENHG(1,3)          ADL01910
      WRITE(RPTOUT,140) KAHG(2,3),EAHG(2,3),KNHG(2,3),ENHG(2,3)          ADL01920
      WRITE(RPTOUT,145) KAHG(3,3),EAHG(3,3),KNHG(3,3),ENHG(3,3)          ADL01930
      WRITE(RPTOUT,150) KBHG(1,3),EBHG(1,3),KOXG(1,3),EOXG(1,3)          ADL01940
      WRITE(RPTOUT,155) KBHG(2,3),EBHG(2,3),KOXG(2,3),EOXG(2,3)          ADL01950
      WRITE(RPTOUT,160) KBHG(3,3),EBHG(3,3),KOXG(3,3),EOXG(3,3)          ADL01960
      WRITE(RPTOUT,175)KBACWG(1,3),QTBAWG(1,3),KBACSG(1,3),QTBASG(1,3)ADL01970
      WRITE(RPTOUT,180)KBACWG(2,3),QTBAWG(2,3),KBACSG(2,3),QTBASG(2,3)ADL01980
      WRITE(RPTOUT,185)KBACWG(3,3),QTBAWG(3,3),KBACSG(3,3),QTBASG(3,3)ADL01990
      WRITE(RPTOUT,190) (ABSG(I,3),I=1,3)                                ADL02000
      WRITE(RPTOUT,195) (ABSG(I,3),I=4,9)                                ADL02010
      WRITE(RPTOUT,195) (ABSG(I,3),I=10,15)                              ADL02020
      WRITE(RPTOUT,195) (ABSG(I,3),I=16,21)                              ADL02030
      WRITE(RPTOUT,195) (ABSG(I,3),I=22,27)                              ADL02040
      WRITE(RPTOUT,195) (ABSG(I,3),I=28,33)                              ADL02050
      WRITE(RPTOUT,195) (ABSG(I,3),I=34,39)                              ADL02060
C-----------------------------------------------------------------------  ADL02070
400   CONTINUE                                                           ADL02080
      IF(SPFLG(4).NE.1) GO TO 500                                        ADL02090
C-----------------------------------------------------------------------  ADL02100
C PRINT DATA FOR SH-                                                     ADL02110
C ------------------                                                     ADL02120
C PAGE HEADER: NAME OF CHEMICAL AND ECOSYSTEM                            ADL02130
      WRITE(RPTOUT,100) CHEMNA,SYSTYP                                    ADL02140
      WRITE(RPTOUT,410)                                                  ADL02150
410   FORMAT(1X,'SH- (SINGLY CHARGED ANION, SPECIES #4) INPUT DATA:')    ADL02160
      WRITE(RPTOUT,415) PKAG(1),EPKAG(1),SOLG(4),ESOLG(4)                ADL02170
415   FORMAT(1X,'PKA1=',1PG11.4,' EPKA1=',G11.4,' SOL=',G11.4,          ADL02180
     2 ' ESOL=',G11.4)                                                   ADL02190
      WRITE(RPTOUT,220) MWTG                                             ADL02200
      WRITE(RPTOUT,425) KPSG(4),KPBG(4),KAECG(1),RFLATG(4)               ADL02210
425   FORMAT(1X,'KPS=',1PG11.4,' KPB=',G11.4,' KAEC=',G11.4             ADL02220
     1,' RFLAT=',G11.4)                                                  ADL02230
      WRITE(RPTOUT,130) KDPG(4),QUANTG(1,4),QUANTG(2,4),QUANTG(3,4)      ADL02240
      WRITE(RPTOUT,135) KAHG(1,4),EAHG(1,4),KNHG(1,4),ENHG(1,4)          ADL02250
      WRITE(RPTOUT,140) KAHG(2,4),EAHG(2,4),KNHG(2,4),ENHG(2,4)          ADL02260
      WRITE(RPTOUT,145) KAHG(3,4),EAHG(3,4),KNHG(3,4),ENHG(3,4)          ADL02270
      WRITE(RPTOUT,150) KBHG(1,4),EBHG(1,4),KOXG(1,4),EOXG(1,4)          ADL02280
      WRITE(RPTOUT,155) KBHG(2,4),EBHG(2,4),KOXG(2,4),EOXG(2,4)
      WRITE(RPTOUT,160) KBHG(3,4),EBHG(3,4),KOXG(3,4),EOXG(3,4)
      WRITE(RPTOUT,175)KBACWG(1,4),QTBAWG(1,4),KBACSG(1,4),QTBASG(1,4)
      WRITE(RPTOUT,180)KBACWG(2,4),QTBAWG(2,4),KBACSG(2,4),QTBASG(2,4)
      WRITE(RPTOUT,185)KBACWG(3,4),QTBAWG(3,4),KBACSG(3,4),QTBASG(3,4)
      WRITE(RPTOUT,190) (ABSG(I,4),I=1,3)
      WRITE(RPTOUT,195) (ABSG(I,4),I=4,9)
      WRITE(RPTOUT,195) (ABSG(I,4),I=10,15)
      WRITE(RPTOUT,195) (ABSG(I,4),I=16,21)
      WRITE(RPTOUT,195) (ABSG(I,4),I=22,27)
      WRITE(RPTOUT,195) (ABSG(I,4),I=28,33)
      WRITE(RPTOUT,195) (ABSG(I,4),I=34,39)
C-----------------------------------------------------------------------
500   CONTINUE
      IF(SPFLG(5).NE.1) GO TO 600
C-----------------------------------------------------------------------
C PRINT DATA FOR S=
C ----------------
C PAGE HEADER: NAME OF CHEMICAL AND ECOSYSTEM
      WRITE(RPTOUT,100) CHEMNA,SYSTYP
      WRITE(RPTOUT,510)
510   FORMAT(1X,'S= (DOUBLY CHARGED ANION, SPECIES #5) INPUT DATA:')
      WRITE(RPTOUT,515) PKAG(2),EPKAG(2),SOLG(5),ESOLG(5)
515   FORMAT(1X,'PKA2=',1PG11.4,' EPKA2=',G11.4,' SOL=',G11.4,
     2 ' ESOL=',G11.4)
      WRITE(RPTOUT,220) MWTG
      WRITE(RPTOUT,425) KPSG(5),KPBG(5),KAECG(2),RFLATG(5)
      WRITE(RPTOUT,130) KDPG(5),QUANTG(1,5),QUANTG(2,5),QUANTG(3,5)
      WRITE(RPTOUT,135) KAHG(1,5),EAHG(1,5),KNHG(1,5),ENHG(1,5)
      WRITE(RPTOUT,140) KAHG(2,5),EAHG(2,5),KNHG(2,5),ENHG(2,5)
      WRITE(RPTOUT,145) KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)
      WRITE(RPTOUT,150) KBHG(1,5),EBHG(1,5),KOXG(1,5),EOXG(1,5)
      WRITE(RPTOUT,155) KBHG(2,5),EBHG(2,5),KOXG(2,5),EOXG(2,5)
      WRITE(RPTOUT,160) KBHG(3,5),EBHG(3,5),KOXG(3,5),EOXG(3,5)
      WRITE(RPTOUT,175)KBACWG(1,5),QTBAWG(1,5),KBACSG(1,5),QTBASG(1,5)
      WRITE(RPTOUT,180)KBACWG(2,5),QTBAWG(2,5),KBACSG(2,5),QTBASG(2,5)
      WRITE(RPTOUT,185)KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)
      WRITE(RPTOUT,190) (ABSG(I,5),I=1,3)
      WRITE(RPTOUT,195) (ABSG(I,5),I=4,9)
      WRITE(RPTOUT,195) (ABSG(I,5),I=10,15)
      WRITE(RPTOUT,195) (ABSG(I,5),I=16,21)
      WRITE(RPTOUT,195) (ABSG(I,5),I=22,27)
      WRITE(RPTOUT,195) (ABSG(I,5),I=28,33)
      WRITE(RPTOUT,195) (ABSG(I,5),I=34,39)
C-----------------------------------------------------------------------
600   CONTINUE
C DATA PRINTING COMPLETE
      RETURN
      END
C FILE 'BLOCK.FTN'
      BLOCK DATA
C
C-----------------------------------------------------------------------
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG       ADL02860
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG  ADL02870
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG     ADL02880
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG     ADL02890
     1,NPSEDG,INTFLG                                                     ADL02900
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH      ADL02910
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                           ADL02920
      LOGICAL*1 CHEMNA,SYSTYP                                            ADL02930
      COMMON/NAME1G/CHEMNA(60)                                           ADL02940
      COMMON/CONTRG/SPFLG(5)                                             ADL02950
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                                ADL02960
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                      ADL02970
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)           ADL02980
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                              ADL02990
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)             ADL03000
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)              ADL03010
     1,KBHG(3,5),EBHG(3,5)                                               ADL03020
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                   ADL03030
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)       ADL03040
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)          ADL03050
     1,IFLLDG(10)                                                        ADL03060
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                 ADL03070
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)    ADL03080
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                   ADL03090
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)     ADL03100
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)      ADL03110
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)               ADL03120
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG          ADL03130
     1,DFACG(10)                                                         ADL03140
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                         ADL03150
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                        ADL03160
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)           ADL03170
     1,INTFLG(10)                                                        ADL03180
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT         ADL03190
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                       ADL03200
C----------------------------------------------------------------------- ADL03210
C FILE 'CHEML.COM'                                                       ADL03220
      COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)             ADL03230
      COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),         ADL03240
     2 VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                      ADL03250
      COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                          ADL03260
      COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                              ADL03270
      COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),              ADL03280
     2 CHEMPC,BIOPCT,VOLPCT,EXPPCT                                       ADL03290
      COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)                  ADL03300
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                          ADL03310
C----------------------------------------------------------------------- ADL03320
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                    ADL03330
C----------------------------------------------------------------------- ADL03340
C FILE 'ENVIRL.COM'                                                      ADL03350
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                     ADL03360
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),          ADL03370
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                             ADL03380
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,             ADL03390
     2 KDTIME,TFACTR,IPER,PERIOD                                         ADL03400
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)          ADL03410
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                        ADL03420
```

```
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),         ADL03430
     1    PSEDCO(10)                                                    ADL03440
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                  ADL03450
     2 SEDFL(10,10),WATINL(10)                                          ADL03460
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),              ADL03470
     1  WAVEL(39),TOTETA(39),WLAML(39)                                  ADL03480
C**********                                                             ADL03490
      COMMON/NPSBLK/NPS,LUNPS,LUNPS2,NPSPER                             ADL03500
      COMMON/DOTBLK/DOT,LUDOT                                           ADL03510
C**********                                                             ADL03520
      REAL INTFL,NPSFL,NPSCOL                                           ADL03530
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                    ADL03540
      LOGICAL PRINT,NPS,DOT                                             ADL03550
      INTEGER L,E,H,B                                                   ADL03560
C**********                                                             ADL03570
      DATA NPS,DOT,LUNPS,LUNPS2,LUDOT,NPSPER/.TRUE.,.TRUE.,9,10,11,1/   ADL03580
C**********                                                             ADL03590
C-----------------------------------------------------------------------ADL03600
C-----------------------------------------------------------------------ADL03610
C DIFFUSE ATTENUATION COEFFICIENTS FOR PURE WATER (DATA FROM SMITH &    ADL03620
C BAKER 1978 (L&O), 1979 (PHOTOCHEM PHOTOBIOL) & ZEPP P.C.):            ADL03630
      DATA WATETA/.198,.186,.175,.166,.157,.150,.142,.136,.129,         ADL03640
     1 .122,.116,.103,.087,.059,.051,.044,.040,.031,.027,.025,.024,     ADL03650
     2 .022,.022,.023,.023,.023,.022,.025,.032,.050,.063,.082,.131,     ADL03660
     3 .244,.336,.485,.630,2.53,2.00/                                   ADL03670
C SPECIFIC DIFFUSE ATTENUATION COEFFICIENTS FOR CHLOROPHYLL PIGMENTS:   ADL03680
C (/METER/(MG PIGMENT/LITER)), DATA FROM SMITH&BAKER 1978)              ADL03690
      DATA PIGETA/13*0.,                                                ADL03700
     1 66.,66.,61.,55.,51.,49.,47.,45.,42.,39.,37.,34.,31.,29.,26.,     ADL03710
     2 23.,17.,12.,9.,7.,6.,9.,16.,4.,2.,0./                            ADL03720
C SPECIFIC DIFFUSE ATTENUATION COEFFICIENTS FOR DISSOLVED ORGANIC       ADL03730
C CARBON (DATA OF R. ZEPP FOR AUCILLA RIVER WATER), (/M/(MG/L))):       ADL03740
      DATA DOCETA/6.3,6.12,5.94,5.76,5.57,5.39,5.22,5.06,4.90,          ADL03750
     1 4.74,4.56,4.17,3.64,3.15,2.74,2.34,2.00,1.64,1.39,1.19,          ADL03760
     2 1.02,.870,.753,.654,.573,.504,.444,.396,.357,.282,.228,.188,     ADL03770
     3 .158,6*0./                                                       ADL03780
C SET UP LETTER CODES FOR DISCRIMINATING AMONG COMPARTMENTS TYPES:      ADL03790
C ( (L)ITTORAL, (E)PILIMNION, (H)YPOLIMNION, (B)ENTHIC                  ADL03800
      DATA L,E,H,B/1HL,1HE,1HH,1HB/                                     ADL03810
C SET LOGICAL UNIT NUMBERS FOR OUTPUT FILES OF BATCH VERSION:           ADL03820
      DATA RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT/                            ADL03830
     1  8,4,3,5,8/                                                      ADL03840
C PROGRAM DIMENSIONS FOR USE AS LOOP TERMINATORS:                       ADL03850
      DATA NPX,NCON/10,30/                                              ADL03860
C                                                                       ADL03870
      DATA BATCH/0/                                                     ADL03880
      END                                                               ADL03890
C FILE 'PRCHEM.FTN'                                                     ADL03900
C FILE 'PRENV.FTN'                                                      ADL03910
      SUBROUTINE PRENV                                                  ADL03920
C---------------------------------------------------------------------- ADL03930
C PRENV. PRENV RECORDS THE CANONICAL CHARACTERIZATION OF THE ECOSYSTEM  ADL03940
C (INPUT DATA) IN THE REPORT FILE ('REPORT.DAT').                       ADL03950
C-----------------------------------------------------------------------ADL03960
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL03970
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL03980
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL03990
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG    ADL04000
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG      ADL04010
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL04020
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL04030
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL04040
     1,NPSEDG,INTFLG                                                     ADL04050
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH      ADL04060
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                           ADL04070
      LOGICAL*1 CHEMNA,SYSTYP                                            ADL04080
      COMMON/NAME1G/CHEMNA(60)                                           ADL04090
      COMMON/CONTRG/SPFLG(5)                                             ADL04100
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                                ADL04110
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                      ADL04120
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)           ADL04130
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                              ADL04140
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)             ADL04150
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)              ADL04160
     1,KBHG(3,5),EBHG(3,5)                                               ADL04170
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                   ADL04180
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)       ADL04190
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)          ADL04200
     1,IFLLDG(10)                                                        ADL04210
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                 ADL04220
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)    ADL04230
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                   ADL04240
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)     ADL04250
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)      ADL04260
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)               ADL04270
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG          ADL04280
     1,DFACG(10)                                                         ADL04290
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                         ADL04300
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                        ADL04310
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)           ADL04320
     1,INTFLG(10)                                                        ADL04330
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT         ADL04340
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                       ADL04350
C----------------------------------------------------------------------- ADL04360
C FILE 'CHEML.COM'                                                       ADL04370
       COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)            ADL04380
       COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),        ADL04390
     2 VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                      ADL04400
       COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                         ADL04410
       COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                             ADL04420
       COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),             ADL04430
     2 CHEMPC,BIOPCT,VOLPCT,EXPPCT                                       ADL04440
       COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)                 ADL04450
       REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                         ADL04460
C----------------------------------------------------------------------- ADL04470
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                    ADL04480
C----------------------------------------------------------------------- ADL04490
C FILE 'ENVIRL.COM'                                                      ADL04500
       COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                    ADL04510
       COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),         ADL04520
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                             ADL04530
       COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,            ADL04540
     2 KDTIME,TFACTR,IPER,PERIOD                                         ADL04550
       COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)         ADL04560
     1,JSAV1,JSAV2,ICALL,NPX,NCON
        COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),
     1     PSEDCO(10)
        COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),
     2 SEDFL(10,10),WATINL(10)
        COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),
     1  WAVEL(39),TOTETA(39),WLAML(39)
        REAL INTFL,NPSFL,NPSCOL
        REAL KO2L,KOUNTW,KOUNTS,LIGHTL
        LOGICAL PRINT
        INTEGER L,E,H,B
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C LOCAL VARIABLES FOR COMPRESSING ADVECTION AND TURBULENCE VECTORS:
C UP TO 30 (3* NO. OF COMPARTMENTS) PAIRINGS PERMITTED:
        DIMENSION LOC1(30),LOC2(30),OUT1(30),OUT2(30)
     1,OUT3(30)
C-----------------------------------------------------------------------
C OBJECT TIME FORMAT TO SUPPRESS PRINTING OF NON-EXISTENT INFLOWS
        DOUBLE PRECISION FMT(9), BLK1, PT1, PT2
        REAL*4 OUT4(6),BLANK
        DATA BLANK/4H    /
        DATA FMT/8H(1X,I2,1,8HX,1A1,2X,6*8H        ,8H)       /
        DATA BLK1,PT2/8H,1A1,10X,8H,G11.4  /
        DATA PT1/8H,1PG11.4/
C FORMAT FOR PAGE HEADER:
1       FORMAT(1H1,'  AERL-ESB MODEL OF FATE OF ORGANIC TOXICANTS',
     11X,'IN AQUATIC ECOSYSTEMS.'/1X,'CHEMICAL: ',60A1/1X,
     2'ECOSYSTEM: ',60A1/1X,72(1H-))
C FORMAT FOR DASHED LINE:
2       FORMAT(1X,72(1H-))
C**********
C
        IF (.NOT.PRINT) RETURN
C
C**********
C TITLE:
        WRITE(RPTOUT,10)
10      FORMAT(1X,'INPUT DATA DESCRIBING ENVIRONMENT: GLOBAL'
     1,' PARAMETERS'/1X,72(1H-))
        WRITE(RPTOUT,20) KOUNT,RAING,CLOUDG,LATG
20      FORMAT(1X,'KOUNT=',I3,4X,' RAIN(MM/MO)=',1PG11.4,
     1' CLOUD(10THS)=',G9.2,' LAT=',0PF6.2)
        WRITE(RPTOUT,30) (WLAMG(I),I=1,3)
30      FORMAT(1X,'IRRADIANCE (WLAM,P/SQ CM/SEC/N NM):',3(1X,1PG11.4))
        WRITE(RPTOUT,40) (WLAMG(I),I=4,9)
        WRITE(RPTOUT,40) (WLAMG(I),I=10,15)
        WRITE(RPTOUT,40) (WLAMG(I),I=16,21)
        WRITE(RPTOUT,40) (WLAMG(I),I=22,27)
        WRITE(RPTOUT,40) (WLAMG(I),I=28,33)
        WRITE(RPTOUT,40) (WLAMG(I),I=34,39)
        WRITE(RPTOUT,2)
40      FORMAT(1X,1PG11.4,5(1X,G11.4))
C-----------------------------------------------------------------------
        WRITE(RPTOUT,1) CHEMNA,SYSTYP
C TABLE HEADER:
        WRITE(RPTOUT,50)
```

```
50        FORMAT(1X,'INPUT DATA DESCRIBING ENVIRONMENT:  BIOLOGICAL '           ADL05140
      1,'PARAMETERS'/1X,72(1H-))                                                ADL05150
          WRITE(RPTOUT,60)                                                      ADL05160
60        FORMAT(1X,' # TY',3X,'BIOMS',7X,'PLRA',6X,'BIOTM',7X,'BACTO'          ADL05170
      1,5X,'ACBAC',8X,'CHL'/10X,'(1)',8X,'(2)',7X,'DEG C.',7X,'(3)',            ADL05180
      2 7X,'(2)',9X,'MG/L')                                                     ADL05190
          WRITE(RPTOUT,2)                                                       ADL05200
          DO 70 J=1,KOUNT                                                       ADL05210
          IF(TYPEE(J).NE.B) WRITE(RPTOUT,80) J,TYPEE(J),BIOMSG(J),              ADL05220
      1 PLRAG(J),BIOTMG(J),BACTOG(J),ACBACG(J),CHLG(J)                          ADL05230
          IF(TYPEE(J).EQ.B) WRITE(RPTOUT,80) J,TYPEE(J),BIOMSG(J),              ADL05240
      1 PLRAG(J),BIOTMG(J),BACTOG(J),ACBACG(J)                                  ADL05250
70        CONTINUE                                                              ADL05260
80        FORMAT(1X,I2,1X,1A1,2X,1PG11.4,5(G11.4))                              ADL05270
          WRITE(RPTOUT,2)                                                       ADL05280
C TABLE FOOTNOTES:                                                              ADL05290
          WRITE(RPTOUT,85)                                                      ADL05300
85        FORMAT(1X,' (1) UNITS: MG/L IN WATER COLUMN (L,H,B); ',               ADL05310
      1'G (D.W.)/SQUARE METER IN (B).'/1X,' (2) DIMENSIONLESS',                 ADL05320
      2' NUMBERS.'/1X,' (3) UNITS: CELLS/ML IN L, E, H; CELLS/100',             ADL05330
      3' G D.W.  IN B.')                                                        ADL05340
C-------------------------------------------------------------------------------ADL05350
          WRITE(RPTOUT,1) CHEMNA,SYSTYP                                         ADL05360
C TABLE HEADER:                                                                 ADL05370
          WRITE(RPTOUT,90)                                                      ADL05380
90        FORMAT(1X,'INPUT DATA DESCRIBING ENVIRONMENT:  DEPTHS AND '           ADL05390
      1,'INFLOWS'/1X,72(1H-))                                                   ADL05400
          WRITE(RPTOUT,95)                                                      ADL05410
95        FORMAT(1X,' # TY',3X,'DEPTH',6X,'STFLO',6X,'STSED',6X,                ADL05420
      1'NPSFL',6X,'NPSED',6X,'INTFL'/11X,'M',7X,'CU M/HR',5X,'KG/HR',           ADL05430
      2 5X,'CU M/HR',5X,'KG/HR',5X,'CU M/HR')                                   ADL05440
          WRITE(RPTOUT,2)                                                       ADL05450
          DO 100 J=1,KOUNT                                                      ADL05460
          OUT4(1)=DEPTHG(J)                                                     ADL05470
          OUT4(2)=STFLOG(J)                                                     ADL05480
          OUT4(3)=STSEDG(J)                                                     ADL05490
          OUT4(4)=NPSFLG(J)                                                     ADL05500
          OUT4(5)=NPSEDG(J)                                                     ADL05510
          OUT4(6)=INTFLG(J)                                                     ADL05520
C BUILD FORMAT STATEMENT:                                                       ADL05530
          KKL=0                                                                 ADL05540
          DO 99 I=1,6                                                           ADL05550
          IF(OUT4(I).NE.0.0) GO TO 97                                           ADL05560
C ZERO VALUE FOR VARIABLE:                                                      ADL05570
          FMT(I+2)=BLK1                                                         ADL05580
          OUT4(I)=BLANK                                                         ADL05590
          GO TO 99                                                              ADL05600
97        CONTINUE                                                              ADL05610
C NON-ZERO VALUE FOR VARIABLE:                                                  ADL05620
          KKL=KKL+1                                                             ADL05630
          FMT(I+2) = PT2                                                        ADL05640
          IF (KKL.EQ.1) FMT(I+2) = PT1                                          ADL05650
99        CONTINUE                                                              ADL05660
          WRITE(RPTOUT,FMT) J, TYPEE(J),OUT4                                    ADL05670
100       CONTINUE                                                              ADL05680
          WRITE(RPTOUT,2)                                                       ADL05690
C-------------------------------------------------------------------------------ADL05700
```

```
      WRITE(RPTOUT,1) CHEMNA,SYSTYP                                     ADL05710
      WRITE(RPTOUT,110)                                                 ADL05720
110   FORMAT(1X,'INPUT DATA DESCRIBING ENVIRONMENT:  SEDIMENT'          ADL05730
     1,' CHARACTERISTICS'/1X,72(1H-))                                   ADL05740
      WRITE(RPTOUT,120)                                                 ADL05750
120   FORMAT(1X,' # TY',3X,'SDCHR',6X,'PCTWA',7X,'FROC',7X,'CEC',       ADL05760
     1 8X,'AEC',7X,'DOC'/10X,'(1)',8X,'(2)',8X,'(3)',8X,                ADL05770
     2'(4)',8X,'(4)',7X,'MG/L')                                         ADL05780
      WRITE(RPTOUT,2)                                                   ADL05790
      DO 130 J=1,KOUNT                                                  ADL05800
      IF(TYPEE(J).NE.B) WRITE(RPTOUT,131) J,TYPEE(J), SDCHRG(J),        ADL05810
     1 FROCG(J),CECG(J),AECG(J),DOCG(J)                                 ADL05820
      IF(TYPEE(J).EQ.B) WRITE(RPTOUT,80) J, TYPEE(J), SDCHRG(J),        ADL05830
     2 PCTWAG(J), FROCG(J), CECG(J), AECG(J)                            ADL05840
130   CONTINUE                                                          ADL05850
131   FORMAT(1X,I2,1X,1A1,2X,1PG11.4,11X,4(G11.4))                      ADL05860
      WRITE(RPTOUT,2)                                                   ADL05870
      WRITE(RPTOUT,135)                                                 ADL05880
135   FORMAT(1X,' (1) UNITS: MG/L SUSPENDED SEDIMENT IN L, E, H;',      ADL05890
     1' BULK DENSITY (G/CC) IN B'/1X,' (2) 100 * F.W./D.W. IN B.'/1X,   ADL05900
     2' (3) DIMENSIONLESS.'/1X,' (4) MEQ/100 GRAMS DRY WEIGHT.')        ADL05910
C ------------------------------------------------------------------------ADL05920
      WRITE(RPTOUT,1) CHEMNA,SYSTYP                                     ADL05930
C TABLE HEADER:                                                         ADL05940
      WRITE(RPTOUT,140)                                                 ADL05950
14C   FORMAT(1X,'INPUT DATA DESCRIBING ENVIRONMENT: AERATION, LIGHT,'   ADL05960
     1,' MISC.'/1X,72(1H-))                                             ADL05970
      WRITE(RPTOUT,150)                                                 ADL05980
150   FORMAT(1X,' # TY',4X,'KO2',7X,'WIND',7X,'CMPET',6X,'DFAC',7X,     ADL05990
     1 'EVAP',8X,'AREA'/7X,'CM/HR @20',2X,'M/S @10 CM',4X,              ADL06000
     2 '/M',8X,'M/M',7X,'MM/MON',7X,'SQ M')                             ADL06010
      WRITE(RPTOUT,2)                                                   ADL06020
      DO 160 J=1,KOUNT                                                  ADL06030
      IF(TYPEE(J).NE.B) WRITE(RPTOUT,80) J,TYPEE(J),KO2G(J),WINDG(J),   ADL06040
     2 CMPETG(J),DFACG(J),EVAPG(J),AREAG(J)                             ADL06050
      IF(TYPEE(J).EQ.B) WRITE(RPTOUT,165) J,TYPEE(J),AREAG(J)           ADL06060
160   CONTINUE                                                          ADL06070
165   FORMAT(1X,I2,1X,1A1,57X,1PG11.4)                                  ADL06080
      WRITE(RPTOUT,2)                                                   ADL06090
C-------------------------------------------------------------------------ADL06100
      WRITE(RPTOUT,1) CHEMNA,SYSTYP                                     ADL06110
      WRITE(RPTOUT,170)                                                 ADL06120
170   FORMAT(1X,'INPUT DATA DESCRIBING ENVIRONMENT: ADVECTIVE'          ADL06130
     1,' INTERCONNECTIONS'/1X,72(1H-))                                  ADL06140
C COMPRESS THE PAIRINGS TO ELIMINATE INACTIVE PAIRS, COUNT THE          ADL06150
C ACTIVE PAIRS:                                                         ADL06160
      KNT=0                                                             ADL06170
C UP TO 30 PAIRINGS MAY BE ACTIVE:                                      ADL06180
      DO 180 I=1,30                                                     ADL06190
      IF (JFRADG(I).EQ.0 .AND. ITOADG(I).EQ.0) GO TO 180                ADL06200
      KNT=KNT+1                                                         ADL06210
      LOC1(KNT)=JFRADG(I)                                               ADL06220
      LOC2(KNT)=ITOADG(I)                                               ADL06230
      OUT1(KNT)=ADVPRG(I)                                               ADL06240
180   CONTINUE                                                          ADL06250
      IF (KNT.LE.0) GO TO 220                                           ADL06260
C PRINT THE ACTIVE PAIRINGS, 5 PER LINE:                                ADL06270
```

```
      J=KNT
      K=1
      IPASS=1
      IF(KNT.GT.5) J=5
 190  CONTINUE
      WRITE(RPTOUT,195) (LOC1(I),I=K,J)
      WRITE(RPTOUT,200) (LOC2(I),I=K,J)
      WRITE(RPTOUT,205) (OUT1(I),I=K,J)
      KNT=KNT-5
      IF(KNT.LE.0) GO TO 240
      IPASS=IPASS+1
      K=5*IPASS-4
      J=J+KNT
      ITEST=IPASS*5
      IF(J.GT.ITEST) J=ITEST
      GO TO 190
 195  FORMAT(1X,'COMP. NO. (JFRAD)',5(4X,I2,5X))
 200  FORMAT(1X,'CONNECTED (ITOAD)',5(4X,I2,5X))
 205  FORMAT(1X,'ADVECTION (ADVPR)',1PG10.3,4(1X,G10.3))
C
C
 220  WRITE(RPTOUT,230)
 230  FORMAT(1X,'NO ADVECTIVE INTERCONNECTIONS FOR THIS SYSTEM.')
C
C-----------------------------------------------------------------------
 240  CONTINUE
      WRITE(RPTOUT,1) CHEMNA,SYSTYP
      WRITE(RPTOUT,250)
 250  FORMAT(1X,'INPUT DATA DESCRIBING ENVIRONMENT: TURBULENT'
     1,' INTERCONNECTIONS')
C COMPRESS THE PAIRINGS TO ELIMINATE INACTIVE PAIRS, COUNT THE
C ACTIVE PAIRS:
      KNT=0
C UP TO 30 PAIRINGS MAY BE AVAILABLE:
      DO 260 I=1,30
      IF(JTURBG(I).EQ.0 .AND. ITURBG(I).EQ.0) GO TO 260
      KNT=KNT+1
      LOC1(KNT)=JTURBG(I)
      LOC2(KNT)=ITURBG(I)
      OUT1(KNT)=XSTURG(I)
      OUT2(KNT)=CHARLG(I)
      OUT3(KNT)=DSPG(I)
 260  CONTINUE
      IF(KNT.LE.0) GO TO 330
      J=KNT
      K=1
      IPASS=1
      IF(KNT.GT.5) J=5
 270  CONTINUE
      WRITE(RPTOUT,280) (LOC1(I),I=K,J)
      WRITE(RPTOUT,290) (LOC2(I),I=K,J)
      WRITE(RPTOUT,300) (OUT1(I),I=K,J)
      WRITE(RPTOUT,310) (OUT2(I),I=K,J)
      WRITE(RPTOUT,320) (OUT3(I),I=K,J)
C
      KNT=KNT-5
      IF(KNT.LE.0) GO TO 350
      IPASS=IPASS+1
      IF(IPASS.EQ.4) WRITE (RPTOUT,1) CHEMNA,SYSTYP                   ADL06850
      IF(IPASS.EQ.4) WRITE (RPTOUT,250)                               ADL06860
      K=5*IPASS-4                                                     ADL06870
      J=J+KNT                                                         ADL06880
      ITEST=IPASS*5                                                   ADL06890
      IF(J.GT.ITEST) J=ITEST                                          ADL06900
      GO TO 270                                                       ADL06910
C                                                                     ADL06920
280   FORMAT('0COMP. NO. (JTURB) ',5(4X,I2,5X))                       ADL06930
290   FORMAT(' CONNECTED (ITURB) ',5(4X,I2,5X))                       ADL06940
300   FORMAT(' X-SECTION (XSTUR) ',1PG10.3,4(1X,G10.3))               ADL06950
310   FORMAT(' CHAR. LN. (CHARL) ',1PG10.3,4(1X,G10.3))               ADL06960
320   FORMAT(' EDDY DISP. (DSP)  ',1PG10.3,4(1X,G10.3))               ADL06970
330   WRITE(RPTOUT,340)                                               ADL06980
340   FORMAT(1X,'NO TURBULENT INTERCONNECTIONS FOR THIS ECOSYSTEM.')  ADL06990
350   CONTINJE                                                        ADL07000
C---------------------------------------------------------------------ADL07010
      RETURN                                                          ADL07020
      END                                                             ADL07030
C FILE 'FLOWS.FTN'                                                    ADL07040
      SUBROUTINE FLOWS                                                ADL07050
      DIMENSION ICPTES(10)                                            ADL07060
C---------------------------------------------------------------------ADL07070
C FLOWS. FLOWS COMPUTES THE ADVECTIVE AND DISPERSIVE TRANSPORT FIELD  ADL07080
C THAT MOVES THE TOXICANT THROUGH THE ECOSYSTEM COMPARTMENTS.  AFTER  ADL07090
C CONVERSION OF THE INPUT HYDROLOGIC DATA TO A FORM SUITABLE FOR FURTHERADL07100
C COMPUTATIONS, FLOWS CALLS 4 SUBSIDIARY ROUTINES: WATADV, SEDADV,    ADL07110
C DISPER, AND PRFLOW.                                                 ADL07120
C---------------------------------------------------------------------ADL07130
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:---ADL07140
       REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG ADL07150
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL07160
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG ADL07170
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG   ADL07180
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL07190
     1,BIOTMG,BACTOG,ACBACG,KG2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG  ADL07200
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG  ADL07210
     1,NPSEDG,INTFLG                                                  ADL07220
       INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH  ADL07230
       INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                       ADL07240
       LOGICAL*1 CHEMNA,SYSTYP                                        ADL07250
       COMMON/NAME1G/CHEMNA(60)                                       ADL07260
       COMMON/CONTRG/SPFLG(5)                                         ADL07270
       COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                            ADL07280
       COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                  ADL07290
       COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)       ADL07300
       COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                          ADL07310
       COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)         ADL07320
       COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)          ADL07330
     1,KBHG(3,5),EBHG(3,5)                                            ADL07340
       COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                               ADL07350
       COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)   ADL07360
       COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)      ADL07370
     1,IFLLDG(10)                                                     ADL07380
       COMMON/NAME2G/SYSTYP(60),TYPEE(10)                             ADL07390
       COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30) ADL07400
                                                                      ADL07410
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL07420
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL07430
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)     ADL07440
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)              ADL07450
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG         ADL07460
     1,DFACG(10)                                                        ADL07470
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                        ADL07480
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                       ADL07490
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)          ADL07500
     1,INTFLG(10)                                                       ADL07510
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL07520
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                      ADL07530
C-----------------------------------------------------------------------ADL07540
C FILE 'CHEML.COM'                                                      ADL07550
      COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)            ADL07560
      COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),        ADL07570
     2 VOLKL(10),EXPOKL(10),TCTKL(10),INTINL(10,10)                     ADL07580
      COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                         ADL07590
      COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                             ADL07600
      COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),             ADL07610
     2 CHEMPC,BIOPCT,VOLPCT,EXPPCT                                      ADL07620
      COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)                 ADL07630
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                         ADL07640
C-----------------------------------------------------------------------ADL07650
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                   ADL07660
C-----------------------------------------------------------------------ADL07670
C FILE 'ENVIRL.COM'                                                     ADL07680
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                    ADL07690
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),         ADL07700
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                            ADL07710
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,            ADL07720
     2 KDTIME,TFACTR,IPER,PERIOD                                        ADL07730
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)         ADL07740
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL07750
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),         ADL07760
     1    PSEDCO(10)                                                    ADL07770
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                  ADL07780
     2 SEDFL(10,10),WATINL(10)                                          ADL07790
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),              ADL07800
     1 WAVEL(39),TOTETA(39),WLAML(39)                                   ADL07810
      REAL INTFL,NPSFL,NPSCOL                                           ADL07820
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                    ADL07830
      LOGICAL PRINT                                                     ADL07840
      INTEGER L,E,H,B                                                   ADL07850
C-----------------------------------------------------------------------ADL07860
C-----------------------------------------------------------------------ADL07870
C PRE-ZERO INDICES:                                                     ADL07880
C UP TO NPX COMPARTMENTS MAY BE ACTIVE:                                 ADL07890
      DO 100 J=1,10                                                     ADL07900
       INDEXW(J)=0                                                      ADL07910
       INDEXS(J)=0                                                      ADL07920
100   CONTINUE                                                          ADL07930
C-----------------------------------------------------------------------ADL07940
C SET UP INDEX VECTORS AND COUNT NUMBER OF ACTIVE WATER COLUMN          ADL07950
C AND BOTTOM SEDIMENT COMPARTMENTS:                                     ADL07960
      KOUNTW=0.                                                         ADL07970
      KOUNTS=0.                                                         ADL07980
      DO 120 J=1,KOUNT                                                  ADL07990
       IF(TYPEE(J).EQ.B) INDEXS(J)=1                                    ADL08000
       IF(TYPEE(J).EQ.L .OR. TYPEE(J).EQ.E .OR. TYPEE(J).EQ.H)          ADL08010
     1  INDEXW(J)=1                                                     ADL08020
       IF(INDEXS(J).EQ.1 .OR. INDEXW(J).EQ.1) GO TO 110                 ADL08030
      WRITE(TTYOUT,130) TYPEE(J)                                        ADL08040
      GO TO 240                                                         ADL08050
110   CONTINUE                                                          ADL08060
       IF(INDEXS(J).EQ.1) KOUNTS=KOUNTS+1.                              ADL08070
       IF(INDEXW(J).EQ.1) KOUNTW=KOUNTW+1.                              ADL08080
120   CONTINUE                                                          ADL08090
130   FORMAT(1X,'COMPARTMENT TYPE "',A1,'" IS NEW TO ME.  PLEASE USE',ADL08100
     1 ' THE'/1X,'STANDARD TYPOLOGY FOR COMPARTMENT TYPES:'/12X,        ADL08110
     2 '"L" FOR LITTORAL,'/12X,'"E" FOR EPILIMNION,'/12X,               ADL08120
     3 '"H" FOR HYPOLIMNION, AND'/12X,'"B" FOR BENTHIC SEDIMENT.'/1X,   ADL08130
     4 '(SIMULATION WAS ABORTED.)')                                     ADL08140
C CONVERSION OF 'GLOBAL.COM' INPUT VARIABLES TO 'ENVL.COM' VARIABLES    ADL08150
C USED IN THE MODEL:                                                    ADL08160
C PRE-ZERO TO ELIMINATE LEFT-OVERS FROM PREVIOUS RUN:                   ADL08170
      DO 140 J=1,KOUNT                                                  ADL08180
       WATVOL(J)=0.                                                     ADL08190
       SEDMSL(J)=0.                                                     ADL08200
       SEDCOL(J)=0.                                                     ADL08210
       RAINFL(J)=0.                                                     ADL08220
       EVAPL(J)=0.                                                      ADL08230
       INTFL(J)=0.                                                      ADL08240
       NPSFL(J)=0.                                                      ADL08250
       STRMFL(J)=0.                                                     ADL08260
       STSCOL(J)=0.                                                     ADL08270
       NPSCOL(J)=0.                                                     ADL08280
140   CONTINUE                                                          ADL08290
C---------------------------------------------------------------------- ADL08300
C SET UP CHECK ON DEFINITION OF ADVECTIVE FLOW FIELD - LOAD TEST VECTOR ADL08310
C WITH 1S FOR COMPARTMENTS THAT INCLUDE AN ACTIVE FLOW PATHWAY          ADL08320
C LEAVING THE COMPARTMENT:                                              ADL08330
      DO 145 J=1,30                                                     ADL08340
        IF(JFRADG(J).LE.0) GO TO 145                                    ADL08350
          JJ=JFRADG(J)                                                  ADL08360
          ICPTES(JJ)=1                                                  ADL08370
145   CONTINUE                                                          ADL08380
C---------------------------------------------------------------------- ADL08390
      TOTIN=0.                                                          ADL08400
      DO 190 J=1,KOUNT                                                  ADL08410
       IF(TYPEE(J).EQ.H .OR. TYPEE(J).EQ.B) GO TO 150                   ADL08420
C CONVERT INPUT RAINFALL AND EVAPORATION (MM/MONTH = LITER/             ADL08430
C SQUARE METER/MONTH) TO LITERS PER HOUR:                               ADL08440
       RAINFL(J)=RAING*AREAG(J)/730.5                                   ADL08450
C                                  730.5 HRS/MONTH                      ADL08460
       EVAPL(J)=EVAPG(J)*AREAG(J)/730.5                                 ADL08470
150   STRMFL(J)=STFLOG(J)*1000.                                         ADL08480
       IF(STRMFL(J).NE.0.) STSCOL(J)=STSEDG(J)/STRMFL(J)                ADL08490
       INTFL(J)=INTFLG(J)*1000.                                         ADL08500
C                             1000. LITERS PER CUBIC METER              ADL08510
      NPSFL(J)=NPSFLG(J)*1000.                                          ADL08520
       IF(NPSFL(J).NE.0.) NPSCOL(J)=NPSEDG(J)/NPSFL(J)                  ADL08530
C                                                                       ADL08540
C NET ADVECTIVE FLOW AVAILABLE TO COMPARTMENT:                          ADL08550
```

```
      WATINL(J)=STRMFL(J)+INTFL(J)+NPSFL(J)+RAINFL(J)-EVAPL(J)
C CHECK FOR PROPER DEFINITION OF ADVECTIVE FLOW FIELD: ANY COMPARTMENT
C WITH A NET ADVECTIVE FLOW MUST INCLUDE A PATHWAY FOR THE FLOW
      IF(WATINL(J).LE.0. .OR. ICPTES(J).NE.0) GO TO 155
C IMPROPER DEFINITION: NET ADVECTIVE FLOW, NO OUTLET - NOTIFY USER AND
C ABORT SIMULATION:
      WRITE(TTYOUT,154) J
      GO TO 240
154   FORMAT(/1X,'HYDROLOGIC DEFINITION OF COMPARTMENT',I4,' IS ',
     1'IMPROPER.'/1X,'THERE IS A NET ADVECTED FLOW LEAVING THIS ',
     2'COMPARTMENT,'/1X,'BUT THE FLOW PATHWAY HAS NOT BEEN SPECIFIED.'/
     31X,'SIMULATION ABORTED.')
C
155   CONTINUE
      TOTIN=TOTIN+WATINL(J)
C
C
C COMPUTE WATER VOLUME AND SEDIMENT MASS IN SYSTEM COMPARTMENTS:
C WATER COLUMN:
      IF(INDEXW(J).EQ.0)  GO TO 160
C CONVERT CUBIC METERS TO LITERS:
      WATVOL(J)=VOLG(J)*1000.
C CONVERT MG/L TO KG/L:
C AN ABSOLUTE ZERO VALUE FOR SUSPENDED SEDIMENTS IS NOT PERMISSABLE
C BECAUSE IT PRODUCES ERRONEOUS CALCULATIONS OF THE
C DISPERSIVE EXCHANGE OF SEDIMENTS BETWEEN THE WATER COLUMN AND THE
C BOTTOM SEDIMENT.  IF AN ABSOLUTE ZERO IS LOADED, IT IS ADJUSTED TO
C BE A NON-ZERO VALUE WITHOUT EFFECT ON THE OTHER COMPUTATIONS:
      IF (SDCHRG(J).EQ.0.) SDCHRG(J)=0.001
      SEDCOL(J)=SDCHRG(J)*1.E-6
C COMPUTE SEDIMENT MASS IN COMPARTMENT:
      SEDMSL(J)=SDCHRG(J)*WATVOL(J)*1.E-6
      GO TO 180
C-----------------------------------------------------------------------
160   CONTINUE
C BOTTOM SEDIMENTS:
C TEST FOR PROPER TREATMENT OF % WATER IN INPUT DATA:
      IF(PCTWAG(J).GE.100.) GO TO 170
      WRITE(TTYOUT,200) J
      GO TO 240
170   CONTINUE
C COMPUTE TOTAL MASS IN COMPARTMENT:
      TOTMAS=SDCHRG(J)*VOLG(J)*1.E3
C             G/CC    * CUBIC MFTERS * 1.E6 CC/CU. M. * 1.E-3 KG/G
C SEDIMENT MASS (KILOGRAMS):
      SEDMSL(J)=TOTMAS/(PCTWAG(J)/100.)
C LITERS OF WATER PRESENT IN COMPARTMENT:
      WATVOL(J)=TOTMAS-SEDMSL(J)
C SEDIMENT CONCENTRATION AS KILOGRAMS/LITER:
      SEDCOL(J)=SEDMSL(J)/WATVOL(J)
180   CONTINUE
190   CONTINUE
200   FORMAT(1X,'THE PERCENT WATER DATA FOR COMPARTMENT ',I4/1X,
     1'IS NOT IN PROPER FORM.  PLEASE RE-ENTER THE DATA AS:'/12X,
     2'100*(FRESH WEIGHT/DRY WEIGHT)'/1X,'WITH WEIGHTS IN THE SAME ',
     3'UNITS.  SIMULATION ABORTED.')
C-----------------------------------------------------------------------
```

```
C CALL SUBROUTINE TO EVALUATE ADVECTIVE FLOWS:                          ADL09130
      CALL WATADV(TOTIN)                                                ADL09140
      IF(IFLAG.EQ.8) GO TO 250                                          ADL09150
C-----------------------------------------------------------------------ADL09160
C CALL SUBROUTINE TO COMPUTE ADVECTIVE SEDIMENT TRANSPORT:              ADL09170
      CALL SEDADV                                                       ADL09180
C-----------------------------------------------------------------------ADL09190
C CALL SUBROUTINE TO COMPUTE DISPERSIVE TRANSPORT OF WATER & SEDIMENT:  ADL09200
      CALL DISPER                                                       ADL09210
C-----------------------------------------------------------------------ADL09220
C CHECK FOR ELEMENTS THAT ARE DISCONNECTED FROM THE REST OF THE         ADL09230
C SYSTEM, ABORT IF NECESSARY SINCE THIS WILL GENERATE ZERO DIVIDES      ADL09240
C LATER IN THE PROGRAM:                                                 ADL09250
C TEST: COLUMN TOTAL + OUTFLOW MUST BE NONZERO                          ADL09260
C     DO 220 J=1,KOUNT                                                  ADL09270
C     TOTOUT=0.                                                         ADL09280
C     TOTOUT=TOTOUT+WATOUL(J)                                           ADL09290
C     DO 210 I=1,KOUNT                                                  ADL09300
C     TOTOUT=TOTOUT+WATFL(I,J)                                          ADL09310
C210   CONTINUE                                                         ADL09320
C     IF(TOTOUT.NE.0.) GO TO 220                                        ADL09330
C     WRITE(TTYOUT,230) J                                               ADL09340
C     GO TO 240                                                         ADL09350
C220   CONTINUE                                                         ADL09360
C230   FORMAT(1X,'SYSTEM DEFINITION ERROR: COMPARTMENT ',I4,/1X,        ADL09370
C    1 'IS NOT CONNECTED TO OTHER SYSTEM ELEMENTS.'/1X,                 ADL09380
C    2 'SIMULATION ABORTED.')                                           ADL09390
C-----------------------------------------------------------------------ADL09400
C PRINT TRANSPORT FLOW FIELD:                                           ADL09410
      CALL PRFLOW                                                       ADL09420
C-----------------------------------------------------------------------ADL09430
      GO TO 250                                                         ADL09440
C SET FLAG WHEN MODEL CANNOT BE RUN:                                    ADL09450
240   IFLAG=8                                                           ADL09460
250   RETURN                                                            ADL09470
      END                                                               ADL09480
C FILE 'WATADV.FTN'                                                     ADL09490
      SUBROUTINE WATADV(TOTIN)                                          ADL09500
C-----------------------------------------------------------------------ADL09510
C WATADV. WATADV USES GAUSSIAN ELIMINATION TO CALCULATE THE TOTAL       ADL09520
C ADVECTION OF WATER AMONG SYSTEM COMPARTMENTS AND THE MAGNITUDE OF NET ADL09530
C ADVECTIVE EXPORTS.                                                    ADL09540
C-----------------------------------------------------------------------ADL09550
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL09560
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL09570
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL09580
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL09590
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPHG,XSTURG,CHARLG,DSPG     ADL09600
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL09610
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL09620
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL09630
     1,NPSEDG,INTFLG                                                    ADL09640
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH     ADL09650
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL09660
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL09670
      COMMON/NAME1G/CHEMNA(60)                                          ADL09680
      COMMON/CONTRG/SPFLG(5)                                            ADL09690
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)
     1,KBHG(3,5),EBHG(3,5)
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)
     1,IFLLDG(10)
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG
     1,DFACG(10)
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)
     1,INTFLG(10)
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:
C-----------------------------------------------------------------------
C FILE 'CHEML.COM'
      COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)
      COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),
     2 VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)
      COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)
      COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV
      COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),
     2 CHEMPC,BIOPCT,VOLPCT,EXPPCT
      COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L
C-----------------------------------------------------------------------
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:
C-----------------------------------------------------------------------
C FILE 'ENVIRL.COM'
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL
      COMMON/RKFL/T, FINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,
     2 KDTIME,TFACTR,IPER,PERIOD
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)
     1,JSAV1,JSAV2,ICALL,NPX,NCON
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),
     1    PSEDCO(10)
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),
     2 SEDFL(10,10),WATINL(10)
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),
     1 WAVEL(39),TOTETA(39),WLAML(39)
      REAL INTFL,NPSFL,NPSCOL
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL
      LOGICAL PRINT
      INTEGER L,E,H,B
```

```
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C DIMENSION MATRIX FOR GAUSSIAN PROCESSOR OF ADVECTIVE FLOW FIELD:
C UP TO 10 COMPARTMENTS PERMITTED IN THIS VERSION:
      DIMENSION AMAT(10,11)
      REAL*8 AMAT
C-----------------------------------------------------------------------
C GENERATE ADVECTIVE FLOW FIELD FROM INPUT DATA:
C  USES VECTORS JFRADG AND ITOADG TO LOAD MATRIX 'WATFL' WITH THE VALUES
C  OF ADVPRG, AND IDENTIFIES ADVPRG FOR COMPARTMENTS THAT ADVECT WATER
C  ACROSS SYSTEM BOUNDARIES (EXPORTS).
C-----------------------------------------------------------------------
C PREZERO THE MATRIX AND VECTOR:
      DO 110 J=1,KOUNT
       WATOUL(J)=0.
       SEDOUL(J)=0.
        DO 100 I=1,KOUNT
         WATFL(I,J)=0.
         SEDFL(I,J)=0.
100     CONTINUE
110   CONTINUE
C-----------------------------------------------------------------------
C LOAD THE MATRIX:
C UP TO 30 ACTIVE PAIRINGS (3* THE MAXIMUM NUMBER OF COMPARTMENTS)
C MAY BE AVAILABLE:
      DO 150 K=1,30
C SIMPLIFY CODE BY LOADING VALUES FROM THE VECTOR TO INDICES:
      J=JFRADG(K)
      I=ITOADG(K)
C CHECK PAIRS:
      IF(J.GE.0 .AND. I.GE.0) GO TO 120
       WRITE(TTYOUT,160)
       GO TO 430
120   IF(.NOT.(J.EQ.0 .AND. I.NE.0)) GO TO 130
       WRITE(TTYOUT,170) I
       GO TO 430
C SKIP (0,0) PAIRS
130   IF(J.EQ.0 .AND. I.EQ.0) GO TO 150
C CHECK FOR ELEMENTS NOT IN THE SYSTEM
      IF(J.LE.KOUNT .AND. I.LE.KOUNT) GO TO 140
       WRITE(TTYOUT,180) J,I,KOUNT
       GO TO 430
C CHECK FOR EXPORT LIST:
140   IF(I.EQ.0) WATOUL(J)=WATOUL(J)+ADVPRG(K)
C LOAD WATFL:
      IF(I.NE.0) WATFL(I,J)=ADVPRG(K)
C
C ISSUE WARNING IF DIAGONAL ELEMENT LOADED:
      IF(J.EQ.I) WRITE(TTYOUT,190) J
150   CONTINUE
160   FORMAT(1X,'A NEGATIVE COMPARTMENT NUMBER WAS ENTERED FOR ',
     1'ADVECTIVE FLOW.'/1X,'SIMULATION ABORTED.')
170   FORMAT(1X,'PLEASE LOAD INPUTS TO THE SYSTEM THROUGH STREAM',
     1 'FLOWS,'/1X,'NON-POINT-SOURCE FLOWS ETC.'/1X,'SIMULATION ',
     2 'ABORTED.  REMOVE THE (0,',I4,') ADVECTIVE PAIR.')
180   FORMAT(1X,'ADVECTION FROM ELEMENT ',I4,' TO ELEMENT ',I4/1X,
     1'IS NOT POSSIBLE - ONLY ',I4,' ARE AVAILABLE.'/1X,
     2 'SYSTEM ABORTED.')
190     FORMAT(1X,'THE SPECIFICATIONS FOR ADVECTIVE FLOWS INCLUDED',
     1' AN INTERNAL'/1X,'RECYCLE ON COMPARTMENT ',I4,'.'/1X,
     2'THIS MAY GENERATE A SYSTEM ABORT.')
C
C ZERO THE MATRIX DIAGONAL JUST IN CASE:
        DC 200 J=1,KOUNT
         WATFL(J,J)=0.
200     CONTINUE
C-----------------------------------------------------------------------
        TOTOUT=0.
        DO 220 J=1,KOUNT
         COLTOT=WATOUL(J)
         TOTOUT=TOTOUT+WATOUL(J)
          DO 210 I=1,KOUNT
           COLTOT=COLTOT+WATFL(I,J)
210       CONTINUE
C
C COLUMNS MUST SUM TO EITHER 1 OR 0 (THE LATTER FOR PURELY DISPERSIVE
C EXCHANGE COMPARTMENTS, E.G. HYPOLIMNION OR BOTTOM SEDIMENTS IN MANY
C CASES).
         IF(COLTOT.EQ.0. .OR. COLTOT.EQ.1.) GO TO 220
         WRITE(TTYOUT,230) J
         GO TO 430
220     CONTINUE
C
230     FORMAT(1X,'SYSTEM DEFINITION ERROR: ADVECTIVE TRANSPORT IN ',
    1' COMPARTMENT'/1X,I4,' DOES NOT CONSERVE MASS.  SIMULATION'
     2 ' ABORTED.')
C-----------------------------------------------------------------------
        IF(TOTOUT.GT.0.) GO TO 250
C CHECK FOR EVAPORATIVE SYSTEM (E.G. POND WITH ONLY EVAPORATIVE
C WATER LOSSES):
        IF(TOTIN .LT. 1.E-4) GO TO 250
C TOTIN IS TOTAL INPUTS TO SYSTEM LESS EVAPORATIVE LOSSES, I.E. THE NET
C FLOW AVAILABLE TO ADVECTIVE PROCESSES.  IF THIS IS NON-ZERO, THE
C SYSTEM DEFINITION IS IMPROPER, BUT SOME ERROR MARGIN IS RETAINED FOR
C THE USER DEFINING AN EVAPORATIVE SYSTEM THROUGH THE CHANGE COMMAND.
        WRITE(TTYOUT,240)
        GO TO 430
C
240     FORMAT(1X,'SYSTEM DEFINITION ERROR - NO OUTLET FLOWS PROVIDED,',
     1 ' EVAPORATION'/1X,'DOES NOT BALANCE THE WATER BUDGET.  ',
     2 'SIMULATION ABORTED.')
C-----------------------------------------------------------------------
250     CONTINUE
C SOLVE SYSTEM FOR TOTAL COMPARTMENT FLOWS BY GAUSSIAN ELIMINATION:
C COPY WATERFLOW MATRIX TO GAUSSIAN PROCESSOR, LOAD 1'S ON DIAGONAL,
C NORMALIZED INFLOWS ON LAST COLUMN:
        DO 270 J=1,KOUNT
         DO 260 I=1,KOUNT
          AMAT(I,J)= -WATFL(I,J)
260      CONTINUE
        AMAT(J,J)=1.0
270     CONTINUE
C
C LOAD TOTAL EXOGENOUS INPUTS ON LAST COLUMN:
      KOUNT1=KOUNT+1
      DO 280 J=1,KOUNT
       AMAT(J,KOUNT1) = 0.00
        IF (TOTIN .NE. 0.) AMAT(J,KOUNT1)=WATINL(J)/TOTIN
280   CONTINUE
C-----------------------------------------------------------------------
C GAUSSIAN ELIMINATION:
      DO 370 I=1,KOUNT
      KEXCH=1
      M=I+1
C TEST OF CURRENT DIAGONAL ELEMENT:
290   IF(DABS(AMAT(I,I)).GT.1.E-05) GO TO 320
C
C INTERCHANGE OF EQUATIONS:
      LL=I+KEXCH
C TEST FOR EXISTENCE OF LL-TH EQUATION:
      IF(LL-KOUNT) 300,300,380
300   DO 310 N=1,KOUNT1
      SAVE=AMAT(I,N)
      AMAT(I,N)=AMAT(LL,N)
      AMAT(LL,N)=SAVE
310   CONTINUE
      KEXCH=KEXCH+1
      GO TO 290
C
C REDUCTION OF ITH EQUATION:
320   DO 330 J=M,KOUNT1
      AMAT(I,J)=AMAT(I,J)/AMAT(I,I)
330   CONTINUE
C ELIMINATION IN OTHER EQUATIONS:
      DO 360 J=1,KOUNT
      IF(J-I) 340,360,340
340   DO 350 K=M,KOUNT1
      AMAT(J,K)=AMAT(J,K)-AMAT(J,I)*AMAT(I,K)
350   CONTINUE
360   CONTINUE
370   CONTINUE
      GO TO 400
C
C SYSTEM FAILURE:
380   CONTINUE
      WRITE(TTYOUT,390)
390   FORMAT(1X,'SYSTEM FAILURE: ADVECTIVE TRANSPORT FIELD'/1X,
     1 'CANNOT BE DEFINED.  SIMULATION ABORTED.')
      GO TO 430
400   CONTINUE
C-----------------------------------------------------------------------
C ADVECTIVE FLOW FIELD COMPLETED:
      DO 420 J=1,KOUNT
C CALCULATE EXPORT FLOWS FROM COMPARTMENT:
      WATOUL(J)=WATOUL(J)*TOTIN*AMAT(J,KOUNT1)
C CALCULATE INTRA-SYSTEM TRANSPORTS:
      DO 410 I=1,KOUNT
      WATFL(I,J)=WATFL(I,J)*TOTIN*AMAT(J,KOUNT1)
410   CONTINUE
420   CONTINUE
C-----------------------------------------------------------------------
```

```
      GO TO 440                                                         ADL11980
430   IFLAG=8                                                           ADL11990
440   RETURN                                                            ADL12000
      END                                                               ADL12010
C FILE 'SEDADV.FTN'                                                     ADL12020
      SUBROUTINE SEDADV                                                 ADL12030
C-----------------------------------------------------------------------ADL12040
C SEDADV. SEDADV USES THE OUTCOME OF WATADV TO COMPUTE THE ENTRAINED    ADL12050
C TRANSPORT OF SUSPENDED SEDIMENTS, SEDIMENT EXPORTS, AND BEDLOADS.     ADL12060
C ----------------------------------------------------------------------ADL12070
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL12080
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL12090
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL12100
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL12110
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG     ADL12120
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL12130
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL12140
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL12150
     1,NPSEDG,INTFLG                                                    ADL12160
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH     ADL12170
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL12180
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL12190
      COMMON/NAME1G/CHEMNA(60)                                          ADL12200
      COMMON/CONTRG/SPFLG(5)                                            ADL12210
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL12220
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL12230
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL12240
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL12250
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL12260
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL12270
     1,KBHG(3,5),EBHG(3,5)                                              ADL12280
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL12290
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL12300
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL12310
     1,IFLLDG(10)                                                       ADL12320
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL12330
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL12340
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL12350
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL12360
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)     ADL12370
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)              ADL12380
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG         ADL12390
     1,DFACG(10)                                                        ADL12400
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                        ADL12410
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                       ADL12420
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)          ADL12430
     1,INTFLG(10)                                                       ADL12440
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL12450
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                      ADL12460
C-----------------------------------------------------------------------ADL12470
C FILE 'CHEML.COM'                                                      ADL12480
        COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)          ADL12490
        COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),      ADL12500
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                    ADL12510
        COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                       ADL12520
        COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                           ADL12530
        COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),           ADL12540
     2 CHEMPC,BIOPCT,VOLPCT,EXPPCT                                      ADL12550
       COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)                ADL12560
       REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                        ADL12570
C-----------------------------------------------------------------------ADL12580
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                   ADL12590
C-----------------------------------------------------------------------ADL12600
C FILE 'ENVIRL.COM'                                                     ADL12610
       COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                   ADL12620
       COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),        ADL12630
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                            ADL12640
       COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,           ADL12650
     2 KDTIME,TFACTR,IPER,PERIOD                                        ADL12660
       COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)        ADL12670
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL12680
       COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),        ADL12690
     1    PSEDCO(10)                                                    ADL12700
       COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                 ADL12710
     2 SEDFL(10,10),WATINL(10)                                          ADL12720
       COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),             ADL12730
     1 WAVEL(39),TOTETA(39),WLAML(39)                                   ADL12740
       REAL INTFL,NPSFL,NPSCOL                                          ADL12750
       REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                   ADL12760
       LOGICAL PRINT                                                    ADL12770
       INTEGER L,E,H,B                                                  ADL12780
C-----------------------------------------------------------------------ADL12790
C-----------------------------------------------------------------------ADL12800
C SEDIMENT ADVECTIVE FLOWS AND EXPORTS:                                 ADL12810
C   THIS TREATMENT OF SEDIMENT FLOWS OF NECESSITY ASSUMES THAT THE      ADL12820
C  COMPARTMENTS HAVE BEEN NUMBERED IN A LOGICAL FASHION.  COMPARTMENT   ADL12830
C  '1' MUST BE A WATER COLUMN.  THE SYSTEM DIAGRAM MUST THEN BE NUMBEREDADL12840
C  IN VERTICAL ORDER, PROCEEDING ALONG THE MAIN ADVECTIVE FLOW PATH OF  ADL12850
C  THE ECOSYSTEM.  FOR EXAMPLE, FOR A RIVER SYSTEM WITH 2 VERTICAL      ADL12860
C  SEDIMENT COMPARTMENTS IN EACH REACH, COMPARTMENT '1' WOULD DESIGNATE ADL12870
C  THE WATER COLUMN OF THE FIRST REACH, '2' THE SHALLOW SEDIMENT OF     ADL12880
C  THE FIRST REACH, '3' ITS DEEPER SEDIMENT, '4' THE WATER COLUMN OF THEADL12890
C  SECOND REACH, '5' ITS SHALLOW SEDIMENT, '6' ITS DEEPER SEDIMENT, '7' ADL12900
C  THE WATER COLUMN OF THE THIRD REACH, ETC.                            ADL12910
C   GIVEN THIS NUMBERING SCHEME, ADVECTIVE FLOWS AMONG SEDIMENT         ADL12920
C  COMPARTMENTS ARE PERMITTED SO THAT BED LOADS CAN MOVE ALONG THE      ADL12930
C  ADVECTIVE FLOW PATHWAY(S) OF THE ECOSYSTEM.  HOWEVER, ADVECTIVE      ADL12940
C  FLOWS FROM THE WATER COLUMN INTO THE SEDIMENT (AND VICE VERSA) ARE   ADL12950
C  TAKEN TO INVOLVE A FLOW OF TOXICANT-LADEN WATER ONLY (AS IN INTERFLOWADL12960
C  AND GROUNDWATER RECHARGE).                                           ADL12970
C   THE SYSTEM SEDIMENTS ARE TREATED AS A NON-CONSERVATIVE CONSTITUENT  ADL12980
C  OF THE ECOSYSTEM, SINCE EITHER SEDIMENT GENESIS OR DEGRADATION       ADL12990
C  (REMINERALIZATION) CAN OCCUR WITHIN THE SYSTEM ELEMENTS.  THUS THE   ADL13000
C  SEDIMENT CONCENTRATIONS IN THE SYSTEM ELEMENTS (LOADED BY THE USER)  ADL13010
C  ARE USED DIRECTLY IN THE COMPUTATIONS, AND NO ATTEMPT IS MADE TO     ADL13020
C  GENERATE A SEDIMENT BUDGET FROM THE INPUT (STREAM, NON-POINT-SOURCE) ADL13030
C  SEDIMENT LOADINGS.  THE LATTER ARE SIMPLY USED TO CONSTRAIN TOXICANT ADL13040
C  LOADINGS TO REASONABLE VALUES (IN SUBROUTINE 'CKLOAD').              ADL13050
C   SINCE THE MODEL PROVIDES FOR EXCHANGES OF SEDIMENT WITH EXTERNAL    ADL13060
C COMPARTMENTS (I.E. BOUNDARY CONDITION DISPERSIVE FLOWS), THE TOTAL    ADL13070
C SEDIMENT BUDGET IS NOT EXPLICITLY DETERMINED.                         ADL13080
C   THUS IN SUMMARY, EXPORTS OF WATER ACROSS SYSTEM BOUNDARIES CARRY AN ADL13090
C  ENTRAINED SEDIMENT FLOW EXCEPT IN THE CASE WHERE THE OUTFLOW IS FROM ADL13100
C  A BOTTOM SEDIMENT COMPARTMENT AND THE (J-1) COMPARTMENT IS ALSO A    ADL13110
```

```
C  SEDIMENT.  INTRA-SYSTEM TRANSPORT OF ENTRAINED SEDIMENT IS SUBJECT TOADL13120
C  THE FOLLOWING CONSTRAINTS:  ADVECTIVE FLOWS BETWEEN WATER COLUMNS    ADL13130
C  CARRY A SEDIMENT LOAD DETERMINED BY THE SEDIMENT CONCENTRATION IN THEADL13140
C  SOURCE COMPARTMENT.  ADVECTIVE FLOWS THAT INVOLVE A SEDIMENT-WATER   ADL13150
C  COLUMN PAIR ARE CONSTRAINED TO A WATER FLOW ONLY (INTERFLOWS AND    ADL13160
C  GROUND-WATER RECHARGE).  ADVECTION BETWEEN SEDIMENTS ENTRAINS A      ADL13170
C  PARALLEL SEDIMENT FLOW ONLY WHEN THE COMPARTMENT NUMBERS INDICATE    ADL13180
C  THAT THE SEDIMENTS ARE NOT VERTICALLY CONNECTED (I.E. THE COMPARTMENTADL13190
C  NUMBERS DIFFER BY MORE THAN ONE DIGIT).                             ADL13200
C--------------------------------------------------------------------- ADL13210
C COMPUTE SEDIMENT EXPORTS AND ENTRAINED INTRASYSTEM FLOWS:            ADL13220
      DO 150 J=1,KOUNT                                                 ADL13230
      IF(INDEXS(J).EQ.1) GO TO 110                                     ADL13240
C SOURCE IS A WATER COLUMN COMPARTMENT:                                ADL13250
C EXPORTS (KG/HR):                                                     ADL13260
      SEDOUL(J)=WATOUL(J)*SEDCOL(J)                                    ADL13270
C ADVECTIVE INTERCHANGE AMONG COMPARTMENTS:                            ADL13280
      DO 100 I=1,KOUNT                                                 ADL13290
C IF THE DESTINATION OF THE FLOW IS A BOTTOM SEDIMENT, ONLY WATER ITSELFADL13300
C IS MOVING; OTHERWISE SEDIMENT IS ENTRAINED IN PROPORTION TO ITS      ADL13310
C CONCENTRATION IN THE SOURCE COMPARTMENT:                             ADL13320
      IF(INDEXS(I).NE.1) SEDFL(I,J)=WATFL(I,J)*SEDCOL(J)               ADL13330
100   CONTINUE                                                         ADL13340
C INCREMENT LOOP TO NEXT COMPARTMENT:                                  ADL13350
      GO TO 150                                                        ADL13360
C                                                                      ADL13370
110   CONTINUE                                                         ADL13380
C BOTTOM SEDIMENT COMPARTMENTS:                                        ADL13390
C                                                                      ADL13400
C IF THIS IS COMPARTMENT ONE, THE USER HAS NOT FOLLOWED THE ASSUMPTIONS ADL13410
C NEEDED FOR THE COMPUTATIONS - ABORT -                                ADL13420
      IF(J-1) 130,120,130                                              ADL13430
120   WRITE(TTYOUT,160)                                                ADL13440
      GO TO 170                                                        ADL13450
130   CONTINUE                                                         ADL13460
C                                                                      ADL13470
C IF THE PREVIOUS COMPARTMENT IS NOT ANOTHER SEDIMENT, EXPORT IS       ADL13480
C TAKEN TO BE A BED LOAD:                                              ADL13490
      IF(INDEXS(J-1).NE.1) SEDOUL(J)=WATOUL(J)*SEDCOL(J)               ADL13500
C   (IF THE PREVIOUS COMPARTMENT IS ANOTHER SEDIMENT, AN EXPORT OF     ADL13510
C    WATER ALONG THIS FLOW PATH IS ASSUMED TO REPRESENT A VERTICAL     ADL13520
C    GROUND-WATER RECHARGE CARRYING A WATER FLOW ONLY.)                ADL13530
C                                                                      ADL13540
C COMPUTE INTRA-SYSTEM SEDIMENT TRANSPORT:                             ADL13550
C                                                                      ADL13560
      DO 140 I=1,KOUNT                                                 ADL13570
C                                                                      ADL13580
C ADVECTIVE SEDIMENT TRANSPORT FROM A SEDIMENT COMPARTMENT TO ANOTHER  ADL13590
C SYSTEM ELEMENT IS PERMITTED ONLY WHEN THE DESTINATION COMPARTMENT IS ADL13600
C ANOTHER SEDIMENT, AND THE COMPARTMENT NUMBER FOR THE OTHER SEDIMENT  ADL13610
C INDICATES THAT THE COMPARTMENTS ARE NOT VERTICALLY ADJACENT.  GIVEN  ADL13620
C THE NUMBERING SCHEME OUTLINED ABOVE AND IN THE USER MANUAL, THIS     ADL13630
C CONSTRAINT ALLOWS ONE TO ENTER BEDLOADS THAT MOVE ALONG THE ADVECTIVE ADL13640
C FLOWPATH(S), BUT LIMITS VERTICAL FLOWS TO WATER-FLOW (INTERFLOW,     ADL13650
C GROUND-WATER RECHARGE) ONLY.                                         ADL13660
      IF(INDEXS(I).EQ.1 .AND. IABS(J-I).GT.1) SEDFL(I,J) =             ADL13670
     1                                        WATFL(I,J)*SEDCOL(J)     ADL13680
```

```
C
140     CONTINUE
C
150     CONTINUE
C
160     FORMAT(1X,'THE SYSTEM ELEMENTS HAVE NOT BEEN PROPERLY',
     1' NUMBERED.'/1X,'COMPARTMENT 1 HAS BEEN DESIGNATED A BOTTOM ',
     2'SEDIMENT.'/1X,'SIMULATION ABORTED.')
C-----------------------------------------------------------------------
        GO TO 180
170     IFLAG=8
180     RETURN
      END
C FILE 'DISPER.FTN'
        SUBROUTINE DISPER
C-----------------------------------------------------------------------
C DISPER. DISPER COMPUTES THE TURBULENT TRANSPORT FIELD (INCLUDING
C BOUNDARY CONDITIONS) FOR WATER AND SEDIMENTS, AND GENERATES THE TOTAL
C TRANSPORT FIELD AS A SUM OF ADVECTIVE AND TURBULENT MOTIONS.
C-----------------------------------------------------------------------
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG
     1,NPSEDG,INTFLG
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT
      LOGICAL*1 CHEMNA,SYSTYP
      COMMON/NAME1G/CHEMNA(60)
      COMMON/CONTRG/SPFLG(5)
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)
     1,KBHG(3,5),EBHG(3,5)
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)
     1,IFLLDG(10)
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG
     1,DFACG(10)
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)
     1,INTFLG(10)
```

```
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL14260
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                      ADL14270
C-----------------------------------------------------------------------ADL14280
C FILE 'CHEML.COM'                                                      ADL14290
      COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)            ADL14300
      COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),        ADL14310
     2 VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                     ADL14320
      COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                         ADL14330
      COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                             ADL14340
      COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),             ADL14350
     2 CHEMPC,BIOPCT,VOLPCT,EXPPCT                                      ADL14360
      COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)                 ADL14370
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                         ADL14380
C-----------------------------------------------------------------------ADL14390
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                   ADL14400
C-----------------------------------------------------------------------ADL14410
C FILE 'ENVIRL.COM'                                                     ADL14420
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                    ADL14430
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),         ADL14440
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                            ADL14450
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,            ADL14460
     2 KDTIME,TFACTR,IPER,PERIOD                                        ADL14470
      COMMON/SETUPL/L,E,H,B,KOUNTW.KOUNTS,INDEXW(10),INDEXS(10)         ADL14480
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL14490
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),         ADL14500
     1   PSEDCO(10)                                                     ADL14510
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                  ADL14520
     2 SEDFL(10,10),WATINL(10)                                          ADL14530
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39).              ADL14540
     1 WAVEL(39),TOTETA(39),WLAML(39)                                   ADL14550
      REAL INTFL,NPSFL,NPSCOL                                           ADL14560
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                    ADL14570
      LOGICAL PRINT                                                     ADL14580
      INTEGER L,E,H,B                                                   ADL14590
C-----------------------------------------------------------------------ADL14600
C-----------------------------------------------------------------------ADL14610
C COMPUTATION OF DISPERSIVE FLOWS:                                      ADL14620
C   IN THIS SERIES OF COMPUTATIONS, THE FLOW FIELD DUE TO DISPERSIVE    ADL14630
C MASS TRANSPORT IS COMPUTED FROM THE USER ENTRIES FOR DISPERSIVE       ADL14640
C COMPARTMENT PAIRINGS (JTURBG AND ITURBG) AND THE CROSS-SECTIONAL      ADL14650
C AREA (SQUARE METERS), CHARACTERISTIC LENGTH (METERS), AND             ADL14660
C EDDY DIFFUSIVITY (SQUARE METERS PER HOUR) SPECIFIED FOR THE EXCHANGE  ADL14670
C PAIRING.  ALL BOUNDARY CONDITIONS FOR DISPERSIVE EXCHANGE ACROSS      ADL14680
C SYSTEM BOUNDARIES ARE TAKEN AS ZERO, SINCE OTHERWISE THE ESTIMATES    ADL14690
C OF PERSISTENCE OF THE TOXICANT (AFTER LOADS ARE REMOVED) BECOME       ADL14700
C INTRACTABLE.                                                          ADL14710
C   THE PROCESSES OCCURRING IN THE WATER COLUMN ARE RATHER DIFFERENT    ADL14720
C THAN THOSE OCCURRING WITHIN THE SEDIMENTS OR AT THE SEDIMENT-WATER    ADL14730
C INTERFACE, EVEN THOUGH THE MATHEMATICAL TREATMENT IS QUITE SIMILAR.   ADL14740
C THE BASIC COMPUTATION ((DSPG * XSTURG) / CHARLG) RESULTS IN A         ADL14750
C VARIABLE WITH DIMENSIONS OF CUBIC METERS PER HOUR, AND CLEARLY REFERS ADL14760
C TO TRANSPORT OF A CUBIC METER OF ENVIRONMENTAL VOLUME.  FOR TRANSPORT ADL14770
C AMONG WATER COLUMN COMPARTMENTS THIS PRESENTS NO DIFFICULTY - THE     ADL14780
C WATER FLOW IS GIVEN DIRECTLY, AND (GIVEN THE NON-CONSERVATIVE         ADL14790
C TREATMENT OF SEDIMENT) THE PARALLEL SEDIMENT FLOWS ARE GIVEN FROM THE ADL14800
C SEDIMENT CONCENTRATIONS IN THE SOURCE COMPARTMENTS.                   ADL14810
C   FOR SEDIMENT-WATER INTERACTIONS, THE DISPERSIVE FLOW IS TAKEN TO    ADL14820
```

```
C REPRESENT (IN EFFECT) A CONTINUAL PROCESS OF SUSPENSION OF THE        ADL14830
C SEDIMENT, EQUILIBRATION WITH THE OVERLYING WATER, AND RESETTLEMENT    ADL14840
C ONTO THE BOTTOM.  THUS THE WATER AND SEDIMENT CONCENTRATIONS (PER     ADL14850
C UNIT ENVIRONMENTAL VOLUME OF THE BOTTOM SEDIMENT) ARE USED TO DEFINE  ADL14860
C A SYMMETRICAL (I.E. MASS BALANCED FOR BOTH WATER AND SEDIMENT)        ADL14870
C EXCHANGE BETWEEN THE BOTTOM SEDIMENT AND THE OVERLYING WATER COLUMN.  ADL14880
C   FOR SEDIMENT-SEDIMENT PAIRINGS, THE AVERAGE CHACTERISTICS (I.E.     ADL14890
C AVERAGE BULK DENSITY AND % WATER) OF THE TWO SEDIMENTS ARE USED TO    ADL14900
C GENERATE A SYMMETRICAL EXCHANGE BETWEEN THE SPECIFIED SEDIMENTS.      ADL14910
C---------------------------------------------------------------------- ADL14920
C CALCULATION OF DISPERSIVE FLUXES:                                     ADL14930
C TAKE PAIRINGS ONE-BY-ONE:                                             ADL14940
C UP TO 30 (3* MAXIMUM NUMBER OF COMPARTMENTS) PAIRINGS MAY BE AVAILABLEADL14950
      DO 180 K=1,30                                                     ADL14960
C INCREMENT LOOP FOR BLANK PAIRINGS:                                    ADL14970
      IF(JTURBG(K).EQ.ITURBG(K) .AND. JTURBG(K).EQ.0) GO TO 180         ADL14980
C ISSUE WARNING FOR INTERNAL PAIRS:                                     ADL14990
      IF(JTURBG(K).NE.ITURBG(K)) GO TO 90                               ADL15000
      WRITE(TTYOUT,190) JTURBG(K)                                       ADL15010
      GO TO 180                                                         ADL15020
C ISSUE WARNING IF ANY DISPERSIVE PAIRING REFERS TO A COMPARTMENT       ADL15030
C NUMBER THAT IS NOT PART OF THE SYSTEM (I.E. .GT. KOUNT)               ADL15040
90    IF (JTURBG(K).LE.KOUNT .AND. ITURBG(K).LE.KOUNT) GO TO 95         ADL15050
C WRITE WARNING AND GO ON:                                              ADL15060
      WRITE(TTYOUT,200) K                                               ADL15070
      GO TO 180                                                         ADL15080
C CHECK PARAMETERS FOR ZERO VALUES:                                     ADL15090
95    IF(DSPG(K).NE.0. .AND. XSTURG(K).NE.0. .AND. CHARLG(K).NE.0.)     ADL15100
     1 GO TO 100                                                        ADL15110
C NOTIFY USER AND SKIP PAIRING:                                         ADL15120
      WRITE(TTYOUT,210) K                                               ADL15130
      GO TO 18C                                                         ADL15140
C-----------------------------------------------------------------------ADL15150
100   CONTINUE                                                          ADL15160
C SIMPLIFY SUBSEQUENT CODE BY ASSIGNING CURRENT COMPARTMENT NUMBERS     ADL15170
C TO INDICES:                                                           ADL15180
      J=JTURBG(K)                                                       ADL15190
      I=ITURBG(K)                                                       ADL15200
C                                                                       ADL15210
C FIND PROPER CASE:                                                     ADL15220
C (BOTH UPSTREAM AND DOWNSTREAM BOUNDARY CONDITIONS ARE CONSIDERED,     ADL15230
C  ALTHOUGH IN THE CURRENT APPLICATION BOTH REFER TO DISPERSIVE         ADL15240
C  EXCHANGES WITH SINKS (I.E. BOUNDARY CONDITION CONCENTRATION = 0.0)   ADL15250
C UPSTREAM BOUNDARY CONDITION:                                          ADL15260
      IF(J.EQ.0) GO TO 140                                              ADL15270
C DOWNSTREAM BOUNDARY CONDITION:                                        ADL15280
      IF(I.EQ.0) GO TO 160                                              ADL15290
C                                                                       ADL15300
C INTERCHANGES AMONG SYSTEM COMPARTMENTS:                               ADL15310
C FIND CASE:                                                            ADL15320
C 1. CHECK FOR BOTTOM SEDIMENT - WATER COLUMN INTERACTION:               ADL15330
      IF(INDEXW(J)-INDEXW(I)) 130,110,130                               ADL15340
C 2. COMPARTMENTS OF SAME TYPE - SEPARATE WATER-WATER FROM SED-SED:     ADL15350
110   CONTINUE                                                          ADL15360
      IF(INDEXS(J).EQ.1) GO TO 120                                      ADL15370
C---------------------------------------------------------------------- ADL15380
C 2A: WATER-COLUMN TO WATER-COLUMN PAIRING:                             ADL15390
```

```
        TEMPL=1000.*DSPG(K)*XSTURG(K)/CHARLG(K)                          ADL15400
        WATFL(I,J)=WATFL(I,J)+TEMPL                                      ADL15410
        SEDFL(I,J)=SEDFL(I,J)+TEMPL*SEDCOL(J)                            ADL15420
        WATFL(J,I)=WATFL(J,I)+TEMPL                                      ADL15430
        SEDFL(J,I)=SEDFL(J,I)+TEMPL*SEDCOL(I)                            ADL15440
        GO TO 180                                                        ADL15450
C----------------------------------------------------------------------- ADL15460
120     CONTINUE                                                         ADL15470
C 2B: SEDIMENT-SEDIMENT PAIRING:                                         ADL15480
        TEMPL=(((WATVOL(J)/VOLG(J))+(WATVOL(I)/VOLG(I)))/2.)*            ADL15490
     1   (DSPG(K)*XSTURG(K)/CHARLG(K))                                   ADL15500
        TEMSED=TEMPL*((SEDCOL(J)+SEDCOL(I))/2.)                          ADL15510
        WATFL(I,J)=WATFL(I,J)+TEMPL                                      ADL15520
        SEDFL(I,J)=SEDFL(I,J)+TEMSED                                     ADL15530
        WATFL(J,I)=WATFL(J,I)+TEMPL                                      ADL15540
        SEDFL(J,I)=SEDFL(J,I)+TEMSED                                     ADL15550
        GO TO 180                                                        ADL15560
C----------------------------------------------------------------------- ADL15570
130     CONTINUE                                                         ADL15580
C 3. INTERNAL DISPERSIVE EXCHANGE BETWEEN WATER COLUMN AND SEDIMENT:     ADL15590
        IF(INDEXS(J).EQ.1) KK=J                                          ADL15600
        IF(INDEXS(I).EQ.1) KK=I                                          ADL15610
        TEMPL=(WATVOL(KK)/VOLG(KK))*DSPG(K)*XSTURG(K)/CHARLG(K)          ADL15620
        TEMSED=TEMPL*SEDCOL(KK)                                          ADL15630
        WATFL(I,J)=WATFL(I,J)+TEMPL                                      ADL15640
        SEDFL(I,J)=SEDFL(I,J)+TEMSED                                     ADL15650
        WATFL(J,I)=WATFL(J,I)+TEMPL                                      ADL15660
        SEDFL(J,I)=SEDFL(J,I)+TEMSED                                     ADL15670
        GO TO 180                                                        ADL15680
C----------------------------------------------------------------------- ADL15690
C ALTHOUGH THERE IS NO DIFFERENCE BETWEEN THE TREATMENT OF UPSTREAM AND  ADL15700
C DOWNSTREAM BOUNDARY CONDITIONS, THE DISTINCTION HAS BEEN RETAINED IN   ADL15710
C THE CODE TO SIMPLIFY MODIFICATIONS.                                    ADL15720
C----------------------------------------------------------------------- ADL15730
140     CONTINUE                                                         ADL15740
C UPSTREAM BOUNDARY CONDITION (J=0):                                     ADL15750
        IF(INDEXS(I).EQ.1) GO TO 150                                     ADL15760
C WATER COLUMN UPSTREAM BOUNDARY CONDITION:                              ADL15770
        TEMPL=1000.*DSPG(K)*XSTURG(K)/CHARLG(K)                          ADL15780
        WATOUL(I)=WATOUL(I)+TEMPL                                        ADL15790
        SEDOUL(I)=SEDOUL(I)+TEMPL*SEDCOL(I)                              ADL15800
        GO TO 180                                                        ADL15810
C----------------------------------------------------------------------- ADL15820
150     CONTINUE                                                         ADL15830
C BOTTOM SEDIMENT UPSTREAM BOUNDARY CONDITION:                           ADL15840
        TEMPL=(WATVOL(I)/VOLG(I))*DSPG(K)*XSTURG(K)/CHARLG(K)            ADL15850
        WATOUL(I)=WATOUL(I)+TEMPL                                        ADL15860
        SEDOUL(I)=SEDOUL(I)+TEMPL*SEDCOL(I)                              ADL15870
        GO TO 180                                                        ADL15880
C----------------------------------------------------------------------- ADL15890
160     CONTINUE                                                         ADL15900
C DOWNSTREAM BOUNDARY CONDITIONS (I=0):                                  ADL15910
        IF(INDEXS(J).EQ.1) GO TO 170                                     ADL15920
C WATER COLUMN DOWNSTREAM BOUNDARY CONDITION:                            ADL15930
        TEMPL=1000.*DSPG(K)*XSTURG(K)/CHARLG(K)                          ADL15940
        WATOUL(J)=WATOUL(J)+TEMPL                                        ADL15950
        SEDOUL(J)=SEDOUL(J)+TEMPL*SEDCOL(J)                              ADL15960
```

```
      GO TO 180                                                         ADL15970
C-----------------------------------------------------------------------ADL15980
170   CONTINUE                                                          ADL15990
C BOTTOM SEDIMENT DOWNSTREAM BOUNDARY CONDITION:                         ADL16000
      TEMPL=(WATVOL(J)/VOLG(J))*DSPG(K)*XSTURG(K)/CHARLG(K)             ADL16010
      WATOUL(J)=WATOUL(J)+TEMPL                                         ADL16020
      SEDOUL(J)=SEDOUL(J)+TEMPL*SEDCOL(J)                               ADL16030
180   CONTINUE                                                          ADL16040
C-----------------------------------------------------------------------ADL16050
C ERROR MESSAGE FORMATS:                                                ADL16060
190   FORMAT(1X,'SYSTEM DEFINITION WARNING:  THE DISPERSIVE PAIRS',     ADL16070
     1' INCLUDE AN INTERNAL'/1X,'MIXING OF COMPARTMENT ',I4,'.  THIS ', ADL16080
     2'PAIR HAS BEEN IGNORED.')                                         ADL16090
200   FORMAT(1X,'SYSTEM DEFINITION WARNING: DISPERSIVE PAIRING',        ADL16100
     1' NUMBER',I4/1X,'REFERS TO A COMPARTMENT THAT IS NOT PART OF',    ADL16110
     2' THIS ECOSYSTEM.'/1X,'THIS SPECIFICATION  HAS BEEN IGNORED.')    ADL16120
210   FORMAT(1X,'SYSTEM DEFINITION WARNING: DISPERSIVE PAIRING ',       ADL16130
     1'NUMBER',I4/1X,'CONTAINS A ZERO AS A PARAMETER VALUE.'/1X,        ADL16140
     2'THIS SPECIFICATION HAS BEEN IGNORED.')                           ADL16150
C-----------------------------------------------------------------------ADL16160
      RETURN                                                            ADL16170
      END                                                               ADL16180
C FILE 'PRFLOW.FTN'                                                     ADL16190
      SUBROUTINE PRFLOW                                                 ADL16200
C-----------------------------------------------------------------------ADL16210
C PRFLOW. PRFLOW WRITES A PROFILE OF THE TRANSPORT CHARACTERISTICS      ADL16220
C OF THE ECOSYSTEM TO THE REPORT FILE ('REPORT.DAT')                    ADL16230
C-----------------------------------------------------------------------ADL16240
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL16250
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL16260
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL16270
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL16280
     1,STRLDG,NPSLDG,PCPLDG,[illegible]LDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG      ADL16290
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL16300
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL16310
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL16320
     1,NPSEDG,INTFLG                                                    ADL16330
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH     ADL16340
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL16350
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL16360
      COMMON/NAME1G/CHEMNA(60)                                          ADL16370
      COMMON/CONTRG/SPFLG(5)                                            ADL16380
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL16390
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL16400
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL16410
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL16420
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL16430
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL16440
     1,KBHG(3,5),EBHG(3,5)                                              ADL16450
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL16460
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL16470
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL16480
     1,IFLLDG(10)                                                       ADL16490
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL16500
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL16510
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL16520
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL16530
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG
     1,DFACG(10)
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)
     1,INTFLG(10)
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:
C-----------------------------------------------------------------------
C FILE 'CHEML.COM'
       COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)
       COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),
     2 VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)
       COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)
       COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV
       COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),
     2 CHEMPC,BIOPCT,VOLPCT,EXPPCT
       COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)
       REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L
C-----------------------------------------------------------------------
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:
C-----------------------------------------------------------------------
C FILE 'ENVIRL.COM'
       COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)
       COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL
       COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,
     2 KDTIME,TFACTR,IPER,PERIOD
       COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)
     1,JSAV1,JSAV2,ICALL,NPX,NCON
       COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),
     1    PSEDCO(10)
       COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),
     2 SEDFL(10,10),WATINL(10)
       COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),
     1 WAVEL(39),TOTETA(39),WLAML(39)
       REAL INTFL,NPSFL,NPSCOL
       REAL KO2L,KOUNTW,KOUNTS,LIGHTL
       LOGICAL PRINT
       INTEGER L,E,H,B
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C SET UP TABLE FOR REPORT:
      IF (PRINT) WRITE(RPTOUT,10)
10    FORMAT(21X,'TRANSPORT PROFILE OF ECOSYSTEM'/1X,72(1H-))
C COLUMN HEADINGS:
      IF (PRINT) WRITE(RPTOUT,20)
20    FORMAT(1X,'CP T*',3X,'VOLUME',4X,'SEDIMENT  WATER FLOW',
     2 2X,'SED. FLOW  RESIDENCE TIME (DAYS)'/4X,'Y',3X,
     3 '(CUBIC M)  MASS (KG) (CU. M/DAY) (KG/DAY)',5X,'WATER',
     4 4X,'SEDIMENTS'/1X,72(1H-))
C-----------------------------------------------------------------------
C COMPUTE AND REPORT TRANSPORT PROFILE:
      DO 100 J=1,KOUNT
      TOTWAT=WATOUL(J)
      TOTSED=SEDOUL(J)
      DO 30 I=1,KOUNT
      TOTWAT=TOTWAT+WATFL(I,J)
      TOTSED=TOTSED+SEDFL(I,J)
30    CONTINUE
C
      TOTWAT=TOTWAT*24.
      TOTSED=TOTSED*24.
C
C RESIDENCE TIMES IN DAYS:
      RTWAT=0.
      IF (TOTWAT.NE.0.) RTWAT=WATVOL(J)/TOTWAT
C SET DUMMY VALUE FOR COMPARTMENTS WITH NO SEDIMENTS:
      RTSED=0.
      IF(TOTSED.NE.0.) RTSED=SEDMSL(J)/TOTSED
C CONVERT WATER FLOW TO CUBIC METERS/DAY:
      TOTWAT=TOTWAT/1000.
C WRITE LINE TO REPORT:
      IF (PRINT)
     $ WRITE(RPTOUT,110) J,TYPEE(J),VOLG(J),SEDMSL(J),TOTWAT,TOTSED,
     2 RTWAT,RTSED
C
100   CONTINUE
C
110   FORMAT(1X,I2,1X,1A1,1X,6(1X,1PG10.3))
C PRINT FOOTNOTE TO TABLE:
      IF (PRINT) WRITE(RPTOUT,120)
120   FORMAT(1X,72(1H-)/1X,'* COMP. TYPE: "L"=LITTORAL; "E"=(EPI) AND',
     2 ' "H"=(HYPO)LIMNION; "B"=BENTHIC')
C-----------------------------------------------------------------------
      RETURN
      END
C FILE 'DISTRB.FTN'
C THIS SUBROUTINE COMPUTES THE FRACTION OF THE TOTAL CONCENTRATION OF
C TOXICANT ('Y' STATE VARIABLE) THAT IS PRESENT IN EACH CHEMICAL SPECIES
C OR FORM.
      SUBROUTINE DISTRB
C-----------------------------------------------------------------------
C DISTRB. DISTRB PARTITIONS THE TOXICANT AMONG ITS IONIC SPECIES
C (UP TO 5), WITH EACH OF THE IONIC SPECIES FURTHER PARTITIONED INTO
C DISSOLVED, SEDIMENT SORBED, AND BIOSORBED FORMS.  THE RESULTING
C DISTRIBUTION COEFFICIENTS ARE USED AS MULTIPLIERS ON THE (INPUT)
C HOMOGENEOUS-PHASE REACTION COEFFICIENTS.
C-----------------------------------------------------------------------
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG
     1,NPSEDG,INTFLG
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT
      LOGICAL*1 CHEMNA,SYSTYP
      COMMON/NAME1G/CHEMNA(60)
      COMMON/CONTRG/SPFLG(5)                                            ADL17680
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL17690
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL17700
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL17710
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL17720
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL17730
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL17740
     1,KBHG(3,5),EBHG(3,5)                                              ADL17750
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL17760
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL17770
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL17780
     1,IFLLDG(10)                                                       ADL17790
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL17800
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL17810
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL17820
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL17830
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)     ADL17840
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)              ADL17850
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG         ADL17860
     1,DFACG(10)                                                        ADL17870
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                        ADL17880
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                       ADL17890
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)          ADL17900
     1,INTFLG(10)                                                       ADL17910
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL17920
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                      ADL17930
C-----------------------------------------------------------------------ADL17940
C FILE 'CHEML.COM'                                                      ADL17950
        COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)          ADL17960
        COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),      ADL17970
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                    ADL17980
        COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                       ADL17990
        COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                           ADL18000
        COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),           ADL18010
     2  CHEMPC,BIOPCT,VOLPCT,EXPPCT                                     ADL18020
        COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)               ADL18030
        REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                       ADL18040
C-----------------------------------------------------------------------ADL18050
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                   ADL18060
C-----------------------------------------------------------------------ADL18070
C FILE 'ENVIRL.COM'                                                     ADL18080
        COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                  ADL18090
        COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),       ADL18100
     2  NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                           ADL18110
        COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,          ADL18120
     2  KDTIME,TFACTR,IPER,PERIOD                                       ADL18130
        COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)       ADL18140
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL18150
        COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),       ADL18160
     1     PSEDCO(10)                                                   ADL18170
        COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                ADL18180
     2 SEDFL(10,10),WATINL(10)                                          ADL18190
        COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),            ADL18200
     1  WAVEL(39),TOTETA(39),WLAML(39)                                  ADL18210
        REAL INTFL,NPSFL,NPSCOL                                         ADL18220
        REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                  ADL18230
        LOGICAL PRINT                                                   ADL18240
      INTEGER L,E,H,B                                                   ADL18250
C-----------------------------------------------------------------------ADL18260
C-----------------------------------------------------------------------ADL18270
      DIMENSION DISFCT(15)                                              ADL18280
C 'DISFCT' ARE CONTRIBUTIONS OF EACH SPECIES (SH2, SH3+, ETC.) TO THE   ADL18290
C DISTRIBUTION CONSTANTS (ALPHA).                                       ADL18300
C-----------------------------------------------------------------------ADL18310
C COMPUTE TOTAL SORBING BIOMASS IN SYSTEM COMPARTMENTS AND THE FRACTION ADL18320
C TRANSPORTED WITH SYSTEM MASS FLOWS.  BIOMSG IS MG/L OF (DRY) BIOMASS  ADL18330
C IN WATER COLUMN COMPARTMENTS; IT MAY INCLUDE MACROPHYTES, BENTHIC     ADL18340
C ALGAE ETC., BUT ALL BIOMASS MUST BE REDUCED TO KG/LITER OF THE        ADL18350
C ASSOCIATED WATER COLUMN.  PLRAG IS THE FRACTION OF THIS BIOMASS THAT  ADL18360
C IS PLANKTONIC, I.E. TRANPORTABLE WITH THE MASS FLOWS THROUGH THE      ADL18370
C COMPARTMENT.  FOR BOTTOM SEDIMENT COMPARTMENTS, WHICH MAY CONTAIN     ADL18380
C PLANT ROOTS AND INFAUNA, BIOMSG IS GRAMS/SQUARE METER (DRY WEIGHT),   ADL18390
C WHICH IS TRANSFORMED HERE TO A FORM THAT CAN BE COMPUTATIONALLY       ADL18400
C RELATED TO (WATER) FLOWS THROUGH THE COMPARTMENT.                     ADL18410
C-----------------------------------------------------------------------ADL18420
C TAKE COMPARTMENTS IN NUMERICAL ORDER:                                 ADL18430
      DO 110 J=1,KOUNT                                                  ADL18440
      IF(INDEXS(J).EQ.1) GO TO 100                                      ADL18450
C WATER COLUMN - CONVERT MG/L TO KG/L                                   ADL18460
      BIOTOL(J)= BIOMSG(J)*1.E-6                                        ADL18470
      GO TO 110                                                         ADL18480
100   CONTINUE                                                          ADL18490
C BOTTOM SEDIMENT COMPARTMENTS: INFRA-BENTHIC BIOMASS EXPRESSED AS      ADL18500
C GRAMS/SQUARE METER OF BOTTOM AREA:                                    ADL18510
C CONVERT VALUES TO KG/LITER OF PORE WATER:                             ADL18520
      BIOTOL(J)=(BIOMSG(J)*AREAG(J)*0.001)/WATVOL(J)                    ADL18530
110   CONTINUE                                                          ADL18540
C-----------------------------------------------------------------------ADL18550
C TRANSFER GLOBAL PARTITION COEFFICIENTS TO LOCAL VALUES:               ADL18560
    DO 130 J=1,KOUNT                                                    ADL18570
      DO 120 I=1,5                                                      ADL18580
        KPSL(I,J)=KPSG(I)                                               ADL18590
120   CONTINUE                                                          ADL18600
130 CONTINUE                                                            ADL18610
C                                                                       ADL18620
C TRANSFER (OR COMPUTE IF APPROPRIATE) KOC:                             ADL18630
      KOCL=KOCG                                                         ADL18640
      IF(KOCG.EQ.0. .AND. KOWG.GT.0.) KOCL=0.63*KOWG                    ADL18650
C FACTOR FOR MULTIPLYING KOW FOR WHOLE SEDIMENT FROM                    ADL18660
C KARICKHOFF, S.W., D.S. BROWN, & T.A. SCOTT. WATER RESEARCH (1979).    ADL18670
C-----------------------------------------------------------------------ADL18680
C TAKE COMPARTMENTS IN NUMERICAL ORDER:                                 ADL18690
      DO 220 J=1,KOUNT                                                  ADL18700
C                                                                       ADL18710
C EVALUATE PARTITION COEFFICIENTS:                                      ADL18720
      IF(KOCL.GT.0.) KPSL(1,J)=KOCL*FROCG(J)                            ADL18730
      IF(KCECG(1).GT.0.) KPSL(2,J)=KCECG(1)*CECG(J)                     ADL18740
      IF(KCECG(2).GT.0.) KPSL(3,J)=KCECG(2)*CECG(J)                     ADL18750
      IF(KAECG(1).GT.0.) KPSL(4,J)=KAECG(1)*AECG(J)                     ADL18760
      IF(KAECG(2).GT.0.) KPSL(5,J)=KAECG(2)*AECG(J)                     ADL18770
C-----------------------------------------------------------------------ADL18780
C ADJUST DISSOCIATION CONSTANTS FOR TEMPERATURE:                        ADL18790
C                                                                       ADL18800
C COMPUTE KELVIN TEMPERATURE (TKEL):                                    ADL18810
```

```
      TKEL=TCELG(J)+273.15
C
C PRE-ZERO THE DISSOCIATION CONSTANTS:
      KB1L(J)=0.
      KB2L(J)=0.
      KA1L(J)=0.
      KA2L(J)=0.
C
C REVALUE ONLY FOR THOSE SPECIES THAT ACTUALLY OCCUR:
      IF(SPFLG(2).EQ.0) GO TO 140
C EVALUATE SH3+
      KB1L(J)=10.**(-PKBG(1))
      IF(EPKBG(1).NE.0.) KB1L(J)=10.**(PKBG(1)-((1000.*EPKBG(1))/
     1 (4.58*TKEL)))
140   CONTINUE
      IF(SPFLG(3).EQ.0) GO TO 150
C EVALUATE SH4++
      KB2L(J)=10.**(-PKBG(2))
      IF(EPKBG(2).NE.0.) KB2L(J)=10.**(PKBG(2)-((1000.*EPKBG(2))/
     1 (4.58*TKEL)))
150   CONTINUE
      IF(SPFLG(4).EQ.0) GO TO 160
C EVALUATE SH-
      KA1L(J)=10.**(-PKAG(1))
      IF(EPKAG(1).NE.0.) KA1L(J)=10.**(PKAG(1)-((1000.*EPKAG(1))/
     1 (4.58*TKEL)))
160   CONTINUE
      IF(SPFLG(5).EQ.0) GO TO 170
C EVALUATE SH=
      KA2L(J)=10.**(-PKAG(2))
      IF(EPKAG(2).NE.0.) KA2L(J)=10.**(PKAG(2)-((1000.*EPKAG(2))/
     1 (4.58*TKEL)))
170   CONTINUE
C DISSOCIATION CONSTANTS NOW ADJUSTED FOR TEMPERATURE.
C-----------------------------------------------------------------------
C COMPUTE LOCAL HYDRONIUM AND HYDROXIDE ION CONCENTRATIONS:
      HPLUS=10.**(-PHG(J))
      HYDRX=10.**(-POHG(J))
C COMPUTE SQUARE OF HYDROGEN ION CONCENTRATION:
      HPLSQ=HPLUS*HPLUS
      HYDSQ=HYDRX*HYDRX
C-----------------------------------------------------------------------
C COMPUTE DISTRIBUTION COEFFICIENTS:
C FIRST COMPUTE SEQUENTIAL CONTRIBUTION OF EACH FACTOR:
C DISSOLVED SH2:
      DISFCT(1)=1.
C SH2 SORBED WITH SEDIMENTS:
      DISFCT(2)=KPSL(1,J)*SEDCOL(J)
C SH2 BIOSORBED:
      DISFCT(3)=KPBG(1)*BIOTOL(J)
C SH3+ DISSOLVED:
      DISFCT(4)=KB1L(J)/HYDRX
C SH3+ SORBED WITH SEDIMENT:
      DISFCT(5)=KPSL(2,J)*KB1L(J)*SEDCOL(J)/HYDRX
C SH3+ BIOSORBED:
      DISFCT(6)=KPBG(2)*KB1L(J)*BIOTOL(J)/HYDRX
C SH4++ DISSOLVED:
```

```
         DISFCT(7)=KB2L(J)*KB1L(J)/HYDSQ                                ADL19390
C SH4++ SORBED WITH SEDIMENT:                                           ADL19400
         DISFCT(8)=KB1L(J)*KB2L(J)*KPSL(3,J)*SEDCOL(J)/HYDSQ            ADL19410
C SH4++ BIOSORBED:                                                      ADL19420
         DISFCT(9)=KB1L(J)*KB2L(J)*KPBG(3)*BIOTOL(J)/HYDSQ              ADL19430
C SH- DISSOLVED:                                                        ADL19440
         DISFCT(10)=KA1L(J)/HPLUS                                       ADL19450
C SH- SORBED WITH SEDIMENT:                                             ADL19460
         DISFCT(11)=KA1L(J)*KPSL(4,J)*SEDCOL(J)/HPLUS                   ADL19470
C SH- BIOSORBED:                                                        ADL19480
         DISFCT(12)=KA1L(J)*KPBG(4)*BIOTOL(J)/HPLUS                     ADL19490
C SH= DISSOLVED:                                                        ADL19500
         DISFCT(13)=KA1L(J)*KA2L(J)/HPLSQ                               ADL19510
C SH= SORBED WITH SEDIMENT:                                             ADL19520
         DISFCT(14)=KA1L(J)*KA2L(J)*KPSL(5,J)*SEDCOL(J)/HPLSQ           ADL19530
C SH= BIOSORBED                                                         ADL19540
         DISFCT(15)=KA1L(J)*KA2L(J)*KPBG(5)*BIOTOL(J)/HPLSQ             ADL19550
C                                                                       ADL19560
C SUM THE FACTORS:                                                      ADL19570
         SUMFCT=0.                                                      ADL19580
         DO 180 I=1,15                                                  ADL19590
          SUMFCT=SUMFCT+DISFCT(I)                                       ADL19600
180      CONTINUE                                                       ADL19610
C                                                                       ADL19620
C COMPUTE THE DISTRIBUTION COEFFICIENTS:                                ADL19630
         DO 190 I=1,15                                                  ADL19640
          ALPHA(I,J)=DISFCT(I)/SUMFCT                                   ADL19650
190      CONTINUE                                                       ADL19660
C---------------------------------------------------------------------- ADL19670
C TOTAL DISSOLVED, SORBED WITH SEDIMENT, AND BIOSORBED TOXICANT:        ADL19680
         DO 210 I=16,18                                                 ADL19690
         ALPHA(I,J)=0.                                                  ADL19700
          K1=I-15                                                       ADL19710
          K2=I-3                                                        ADL19720
           DO 200 K=K1,K2,3                                             ADL19730
               ALPHA(I,J)=ALPHA(I,J)+ALPHA(K,J)                         ADL19740
200        CONTINUE                                                     ADL19750
210      CONTINUE                                                       ADL19760
C---------------------------------------------------------------------- ADL19770
C FOR THOSE CASES WITH ZERO SUSPENDED SEDIMENTS, ALPHA(17,J) WILL       ADL19780
C NOW SERVE TO ZERO OUT SEDIMENT-SORBED TRANSPORT OF TOXICANT.          ADL19790
C THEREFORE SEDCOL (SEDIMENT CONCENTRATIONS) WILL NOW BE SET WITH       ADL19800
C DUMMY UNIT VALUES (WHERE NECESSARY) IN ORDER TO PREVENT DIVISION      ADL19810
C BY ZERO IN BOTH THE TRANSPORT COMPUTATIONS AND THE OUTPUT SUBROUTINES.ADL19820
         IF (SEDCOL(J).EQ.0.) SEDCOL(J)=1.                              ADL19830
C---------------------------------------------------------------------- ADL19840
C THE SAME CONSIDERATIONS APPLY FOR THE BIOMASS VARIABLE:               ADL19850
         IF (BIOTOL(J).EQ.0.) BIOTOL(J)=1.                              ADL19860
C-----------------------------------------------------------------------ADL19870
C END OF COMPARTMENT LOOP:                                              ADL19880
220      CONTINUE                                                       ADL19890
C---------------------------------------------------------------------- ADL19900
         RETURN                                                         ADL19910
         END                                                            ADL19920
C FILE 'CKLOAD.FTN'                                                     ADL19930
         SUBROUTINE CKLOAD                                              ADL19940
C---------------------------------------------------------------------- ADL19950
```

```
C CKLOAD. SUBROUTINE CKLOAD ACCUMULATES THE TOTAL LOADINGS TO SYSTEM    ADL19960
C ELEMENTS, CHECKS TO ENSURE THAT NONE OF THE LOADINGS (VIA RAINFALL,   ADL19970
C INTERFLOW, POINT (STREAM) OR NON-POINT SOURCES) ARE AT AQUEOUS        ADL19980
C CONCENTRATIONS (AFTER ACCOUNTING FOR PARTITIONING IN THE INPUT        ADL19990
C FLOWS) GREATER THE 80% OF THE AQUEOUS SOLUBILITY AT THE TEMPERATURE   ADL20000
C SPECIFIED FOR THE SYSTEM ELEMENT LOADED BY THE FLOW.  (THIS CONSTRAINTADL20010
C IS IMPOSED TO MAINTAIN THE LINEAR SORPTION ISOTHERMS USED IN THE MODELADL20020
C WITHIN THEIR PROPER RANGE AND TO PREVENT LOADINGS OF SOLID TOXICANT   ADL20030
C PARTICLES.)                                                           ADL20040
C-----------------------------------------------------------------------ADL20050
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL20060
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL20070
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL20080
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG     ADL20090
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL20100
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL20110
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL20120
     1,NPSEDG, INTFLG                                                   ADL20130
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH     ADL20140
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL20150
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL20160
      COMMON/NAME1G/CHEMNA(60)                                          ADL20170
      COMMON/CONTRG/SPFLG(5)                                            ADL20180
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL20190
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL20200
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL20210
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL20220
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL20230
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL20240
     1,KBHG(3,5),EBHG(3,5)                                              ADL20250
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL20260
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL20270
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL20280
     1,IFLLDG(10)                                                       ADL20290
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL20300
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL20310
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL20320
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL20330
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)     ADL20340
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)              ADL20350
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG         ADL20360
     1,DFACG(10)                                                        ADL20370
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                        ADL20380
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                       ADL20390
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)          ADL20400
     1,INTFLG(10)                                                       ADL20410
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL20420
C-----------------------------------------------------------------------ADL20430
C FILE 'CHEML.COM'                                                      ADL20440
        COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)          ADL20450
        COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),      ADL20460
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                    ADL20470
        COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                       ADL20480
        COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                           ADL20490
        COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),           ADL20500
     2  CHEMPC,BIOPCT,VOLPCT,EXPPCT                                     ADL20510
        COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)               ADL20520
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C FILE 'ENVIRL.COM'
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,
     2 KDTIME,TFACTR,IPER,PERIOD
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)
     1,JSAV1,JSAV2,ICALL,NPX,NCON
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),
     1    PSEDCO(10)
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),
     2SEDFL(10,10),WATINL(10)
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),
     1 WAVEL(39),TOTETA(39),WLAML(39)
C**********
     COMMON/NPSBLK/NPS,LUNPS,LUNPS2,NPSPER
     COMMON/DOTBLK/DOT,LUDOT
C**********
      REAL INTFL,NPSFL,NPSCOL
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL
      INTEGER L,E,H,B
      LOGICAL PRINT,DOT,NPS
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C LOCAL VARIABLES:
      DIMENSION BETA(5),DISFCT(10),CHECK(5)
C BETA ARE FRACTIONS OF TOTAL TOXICANT PRESENT IN THE 5 DISSOLVED
C SPECIES, DISFCT ARE ELEMENTS OF THE DISTRIBUTION FUNCTIONS,
C CHECK VALUES ARE 80% OF SOLUBILITIES FOR CHECKING THAT LOADS
C DO NOT EXCEED SOLUBILITY LIMITS.
C-----------------------------------------------------------------------
C FLAG TO SUPPRESS MULTIPLE PRINTINGS OF ION SOLUBILITY WARNING (S120)
      IPRFLG=0
C COMPUTE SOLUBILITIES:
      DO 110 J=1,KOUNT
      TKEL=273.15+TCELG(J)
C COMPUTE SOLUBILITY OF THE 5 SPECIES IN THIS COMPARTMENT:
      DO 100 I=1,5
C INCREMENT LOOP IF SPECIES I DOES NOT OCCUR:
      IF(SPFLG(I).EQ.0) GO TO 100
C ACQUIRE SOLUBILITY:
      YSATL(I,J)=SOLG(I)
C CHECK FOR ARRHENIUS FUNCTION:
      IF(ESOLG(I) .EQ. 0.) GO TO 90
C SOLUBILITY DATA LOADED AS ARRHENIUS FUNCTION --
C
C CHECK FOR INCONSISTENT INPUT DATA: ARRHENIUS FUNCTION LACKING MWTG -
      IF(MWTG .GT. 0.) GO TO 85
C IMPROPER INPUT DATA:
      IFLAG=8
      WRITE(TTYOUT,80)
      GO TO 420
80    FORMAT(1X,'THE INPUT CHEMICAL DATA CONTAINS AN INCONSISTENCY ',
     1'THAT'/1X,'MUST BE CORRECTED BEFORE EXAMS CAN RUN:'/1X,
     2'THE MOLAR SOLUBILITY OF THE  COMPOUND IS DESCRIBED VIA AN',      ADL21100
     3' ARRHENIUS'/1X,'FUNCTION, BUT THE DATA BASE INDICATES THAT THE'  ADL21110
     4/1X,'MOLECULAR WEIGHT OF THE COMPOUND IS ZERO!')                  ADL21120
C                                                                       ADL21130
C                                                                       ADL21140
85      CONTINUE                                                        ADL21150
C ADJUST FOR LOCAL TEMPERATURE:                                         ADL21160
        YSATL(I,J)=(10.**(SOLG(I)-((1000.*                              ADL21170
     1 ESOLG(I))/(4.58*TKEL))))*MWTG*1000.                              ADL21180
C ARRHENIUS SOLUBILITY IS MOLAR, MULTIPLY BY GRAM MOLECULAR WEIGHT      ADL21190
C AND 1000 MG/G TO CONVERT TO MG/LITER.                                 ADL21200
C                                                                       ADL21210
90      CONTINUE                                                        ADL21220
C ESTIMATE ION SOLUBILITY AS 5000.* NEUTRAL SPECIES SOLUBILITY IF       ADL21230
C USER NEGLECTED TO LOAD A VALUE:                                       ADL21240
        IF(IPRFLG.GT.0 .OR. I.EQ.1 .OR. YSATL(I,J).GT.0.) GO TO 100     ADL21250
        WRITE(TTYOUT,120)                                               ADL21260
        YSATL(I,J)=5000.*YSATL(1,J)                                     ADL21270
        IPRFLG=1                                                        ADL21280
100     CONTINUE                                                        ADL21290
110     CONTINUE                                                        ADL21300
120     FORMAT(1X,'SOLUBILITY OF IONIC SPECIES ESTIMATED FROM THAT OF', ADL21310
     1 ' THE NEUTRAL MOLECULE,'/1X,'SINCE THIS INFORMATION WAS '        ADL21320
     2 'OMITTED FROM THE INPUT DATA.')                                  ADL21330
C---------------------------------------------------------------------- ADL21340
C EVALUATE VALIDITY OF LOADS AND CORRECT THOSE THAT EXCEED 80% OF       ADL21350
C AQUEOUS SOLUBILITY - THE PROCEDURE USES THE TEMPERATURE AND PH OF     ADL21360
C THE TARGET COMPARTMENT IN ITS COMPUTATIONS.                           ADL21370
C---------------------------------------------------------------------- ADL21380
C TAKE ELEMENTS IN ORDER:                                               ADL21390
        DO 390 J=1,KOUNT                                                ADL21400
C INCREMENT LOOP IF THIS COMPARTMENT RECEIVES NO LOADINGS:              ADL21410
        TOTLDL(J)=STRLDG(J)+NPSLDG(J)+PCPLDG(J)+DRFLDG(J)+IFLLDG(J)     ADL21420
        IF(TOTLDL(J).EQ.0.) GO TO 390                                   ADL21430
C COMPUTE TEST VALUES FOR SOLUBILITY CHECKS:                            ADL21440
        DO 130 I=1,5                                                    ADL21450
         CHECK(I)=0.80*YSATL(I,J)                                       ADL21460
130     CONTINUE                                                        ADL21470
C                                                                       ADL21480
C COMPUTE H+ AND OH- FOR PARTITIONING COMPUTATIONS:                     ADL21490
        HPLUS=10.**(-PHG(J))                                            ADL21500
        HYDRX=10.**(-POHG(J))                                           ADL21510
        HPLSQ=HPLUS*HPLUS                                               ADL21520
        HYDSQ=HYDRX*HYDRX                                               ADL21530
C COMPUTE FIRST 5 DISTRIBUTION FACTORS:                                 ADL21540
        DISFCT(1)=1.                                                    ADL21550
        DISFCT(2)=KB1L(J)/HYDRX                                         ADL21560
        DISFCT(3)=KB1L(J)*KB2L(J)/HYDSQ                                 ADL21570
        DISFCT(4)=KA1L(J)/HPLUS                                         ADL21580
        DISFCT(5)=KA1L(J)*KA2L(J)/HPLSQ                                 ADL21590
C                                                                       ADL21600
C TAKE LOADING TYPES ONE BY ONE:                                        ADL21610
C IF NO RAINFALL OR INTERFLOW LOADS PRESENT, SKIP TO NPS & STREAM LOADS:ADL21620
        IF(IFLLDG(J).EQ.0. .AND. PCPLDG(J).EQ.0.) GO TO 250             ADL21630
C                                                                       ADL21640
C CALCULATE DISTRIBUTION COEFFICIENTS FOR INTERFLOW AND RAINFALL:       ADL21650
        SUMFCT=0.                                                       ADL21660
```

```
      DO 140 I=1,5                                                      ADL21670
      SUMFCT=SUMFCT+DISFCT(I)                                           ADL21680
140   CONTINUE                                                          ADL21690
      DO 150 I=1,5                                                      ADL21700
      BETA(I)=DISFCT(I)/SUMFCT                                          ADL21710
150   CONTINUE                                                          ADL21720
C ----------------------------------                                    ADL21730
C 1. INTERFLOW LOAD: WATER FLOW ONLY                                    ADL21740
C-----------------------------------                                    ADL21750
C IF NO LOAD PRESENT, GO ON TO RAINFALL                                 ADL21760
      IF(IFLLDG(J).EQ.0.) GO TO 200                                     ADL21770
C                                                                       ADL21780
C IF THERE IS A LOAD BUT THE INTERFLOW ITSELF IS ZERO:                  ADL21790
      IF(INTFL(J).EQ.0.) GO TO 180                                      ADL21800
C COMPUTE TOTAL CONCENTRATION (MG/L) IN INTERFLOW:                      ADL21810
      TOTCON=1.E6*IFLLDG(J)/INTFL(J)                                    ADL21820
C                                                                       ADL21830
C TAKE CHEMICAL SPECIES IN ORDER:                                       ADL21840
      DO 160 I=1,5                                                      ADL21850
C                                                                       ADL21860
C INCREMENT SPECIES LOOP IF SPECIES DOES NOT OCCUR:                     ADL21870
      IF(SPFLG(I).EQ.0) GO TO 160                                       ADL21880
C                                                                       ADL21890
C COMPUTE ACTUAL CONCENTRATION OF SPECIES:                              ADL21900
      SATTST=BETA(I)*TOTCON                                             ADL21910
C                                                                       ADL21920
      IF(SATTST.LE.CHECK(I)) GO TO 160                                  ADL21930
C                                                                       ADL21940
C SOLUBILITY LIMIT EXCEEDED - MODIFY THE LOAD AND NOTIFY USER:          ADL21950
      IFLLDG(J)=CHECK(I)*INTFL(J)/1.E6                                  ADL21960
      WRITE(TTYOUT,170) J                                               ADL21970
C                                                                       ADL21980
160   CONTINUE                                                          ADL21990
C                                                                       ADL22000
170   FORMAT(1X,'INTERFLOW LOADING EXCEEDED SOLUBILITY LIMIT IN ',      ADL22010
     1  'ELEMENT ',I2,'.'/1X,'LOAD HAS BEEN ADJUSTED.')                 ADL22020
C                                                                       ADL22030
      GO TO 200                                                         ADL22040
C                                                                       ADL22050
180   CONTINUE                                                          ADL22060
C NO INTERFLOW TO THIS COMPARTMENT:                                     ADL22070
      IFLLDG(J)=0.                                                      ADL22080
      WRITE(TTYOUT,190) J                                               ADL22090
190   FORMAT(1X,'AN INTERFLOW LOAD WAS ENTERED FOR ELEMENT ',I2,        ADL22100
     1  ','/1X,'WHICH RECEIVES NO INTERFLOW.  THE LOAD HAS BEEN ',      ADL22110
     2  'REMOVED.')                                                     ADL22120
200   CONTINUE                                                          ADL22130
C---------------------------------------------------------------+      ADL22140
C 2. RAINFALL LOADINGS: ACCOUNTING FOR SUSPENDED SOLIDS           |      ADL22150
C---------------------------------------------------------------+      ADL22160
C IF NO LOAD, GO ON:                                                    ADL22170
      IF(PCPLDG(J).EQ.0.) GO TO 250                                     ADL22180
C IF LOAD, BUT NO RAINFALL:                                             ADL22190
      IF(RAINFL(J).EQ.0.) GO TO 230                                     ADL22200
C COMPUTE TOTAL CONCENTRATION IN RAINFALL:                              ADL22210
      TOTCON=1.E6*PCPLDG(J)/RAINFL(J)                                   ADL22220
C                                                                       ADL22230
```

```
C                                                                       ADL22240
C=======================================================================ADL22250
C                                                                       ADL22260
C---  CALCULATE DISTRIBUTION COEFFICIENTS..                             ADL22270
      DISFCT(6) = KPSL(1,J) * PSEDCO(J)                                 ADL22280
      DISFCT(7) = KPSL(2,J) * PSEDCO(J) * KB1L(J) / HYDRX               ADL22290
      DISFCT(8) = KPSL(3,J) * PSEDCO(J) * KB1L(J) * KB2L(J) / HYDSQ     ADL22300
      DISFCT(9) = KPSL(4,J) * PSEDCO(J) * KA1L(J) / HPLUS               ADL22310
      DISFCT(10)= KPSL(5,J) * PSEDCO(J) * KA1L(J) * KA2L(J) / HPLSQ     ADL22320
C                                                                       ADL22330
      SUMFCT = 0.0                                                      ADL22340
      DO 202 K = 1,10                                                   ADL22350
      SUMFCT = SUMFCT + DISFCT(K)                                       ADL22360
202   CONTINUE                                                          ADL22370
C                                                                       ADL22380
      DO 205 K = 1,5                                                    ADL22390
      BETA(K) = DISFCT(K) / SUMFCT                                      ADL22400
205   CONTINUE                                                          ADL22410
C                                                                       ADL22420
C=======================================================================ADL22430
C                                                                       ADL22440
        DO 210 I=1,5                                                    ADL22450
        IF(SPFLG(I).EQ.0) GO TO 210                                     ADL22460
C ACTUAL CONCENTRATION                                                  ADL22470
        SATTST=BETA(I)*TOTCON                                           ADL22480
        IF(SATTST.LE.CHECK(I))GO TO 210                                 ADL22490
C                                                                       ADL22500
C SOLUBILITY LIMIT EXCEEDED - MODIFY LOAD AND NOTIFY USER               ADL22510
        PCPLDG(J)=CHECK(I)*RAINFL(J)/1.E6                               ADL22520
        WRITE(TTYOUT,220) J                                             ADL22530
210     CONTINUE                                                        ADL22540
220     FORMAT(1X,'RAINFALL LOADING EXCEEDED SOLUBILITY LIMIT IN '      ADL22550
     1 'ELEMENT ',I2,'.'/1X,'LOAD HAS BEEN ADJUSTED.')                  ADL22560
        GO TO 250                                                       ADL22570
230     CONTINUE                                                        ADL22580
C NO RAINFALL:                                                          ADL22590
        PCPLDG(J)=0.                                                    ADL22600
        WRITE(TTYOUT,240) J                                             ADL22610
240     FORMAT(1X,'A RAINFALL LOADING WAS ENTERED FOR ELEMENT ',I2,     ADL22620
     1  ','/1X,'WHICH GETS NO RAIN.  THE LOAD HAS BEEN ',               ADL22630
     2  'REMOVED.')                                                     ADL22640
250     CONTINUE                                                        ADL22650
C---------------------------------------------------------------------- ADL22660
C STREAM AND NPS LOADINGS:                                              ADL22670
C IF BOTH ZERO, GO ON:                                                  ADL22680
        IF(STRLDG(J).EQ.0. .AND. NPSLDG(J).EQ.0.) GO TO 390             ADL22690
C---------------------------------------------------------------------- ADL22700
C-----------------                                                      ADL22710
C 3. STREAM LOADS:                                                      ADL22720
C-----------------                                                      ADL22730
C IF ZERO, GO ON                                                        ADL22740
        IF(STRLDG(J).EQ.0.) GO TO 320                                   ADL22750
C IF A LOAD IS PRESENT WITHOUT AN ASSOCIATED CARRIER FLOW:              ADL22760
        IF(STRMFL(J).EQ.0.) GO TO 300                                   ADL22770
C TOTAL CONCENTRATION:                                                  ADL22780
        TOTCON=1.E6*STRLDG(J)/STRMFL(J)                                 ADL22790
C COMPUTE DISTRIBUTION FACTORS FOR STREAM:                              ADL22800
```

```
      DISFCT(6)=KPSL(1,J)*STSCOL(J)
      DISFCT(7)=KPSL(2,J)*KB1L(J)*STSCOL(J)/HYDRX
      DISFCT(8)=KPSL(3,J)*KB1L(J)*KB2L(J)*STSCOL(J)/HYDSQ
      DISFCT(9)=KA1L(J)*KPSL(4,J)*STSCOL(J)/HPLUS
      DISFCT(10)=KA1L(J)*KA2L(J)*KPSL(5,J)*STSCOL(J)/HPLSQ
C
      SUMFCT=0.
      DO 260 K=1,10
      SUMFCT=SUMFCT+DISFCT(K)
260   CONTINUE
C
C CALCULATE DISTRIBUTION COEFFICIENTS:
C
      DO 270 K=1,5
      BETA(K)=DISFCT(K)/SUMFCT
270   CONTINUE
C
      DO 280 I=1,5
      IF(SPFLG(I).EQ.0) GO TO 280
      SATTST=BETA(I)*TOTCON
      IF(SATTST.LE.CHECK(I)) GO TO 280
      STRLDG(J)=CHECK(I)*STRMFL(J)/1.E6
      WRITE(TTYOUT,290) J
280   CONTINUE
290   FORMAT(1X,'STREAM LOADING EXCEEDED SOLUBILITY LIMIT IN ',
     1 'ELEMENT ',I2,'.'/1X,'LOAD HAS BEEN ADJUSTED.')
      GO TO 320
300   CONTINUE
      STRLDG(J)=0.
      WRITE(TTYOUT,310) J
310   FORMAT(1X,'A STREAMFLOW LOADING WAS ENTERED FOR ELEMENT ',I2,
     1  ','/1X,'WHICH RECEIVES NO STREAM FLOW.  THE LOAD HAS BEEN ',
     2  'REMOVED.')
C
320   CONTINUE
C-------------------------
C 4. NON-POINT-SOURCE LOADS:
C-------------------------
C
      IF((NPSLDG(J).EQ.0.).OR.NPS) GO TO 390
C
      IF(NPSFL(J).EQ.0.) GO TO 370
      TOTCON=1.E6*NPSLDG(J)/NPSFL(J)
C
C CALCULATE DISTRIBUTION COEFFICIENTS:
      DISFCT(6)=KPSL(1,J)*NPSCOL(J)
      DISFCT(7)=KPSL(2,J)*NPSCOL(J)*KB1L(J)/HYDRX
      DISFCT(8)=KPSL(3,J)*NPSCOL(J)*KB1L(J)*KB2L(J)/HYDSQ
      DISFCT(9)=KPSL(4,J)*NPSCOL(J)*KA1L(J)/HPLUS
     DISFCT(10)=KPSL(5,J)*NPSCOL(J)*KA1L(J)*KA2L(J)/HPLSQ
C
      SUMFCT=0.
      DO 330 K=1,10
      SUMFCT=SUMFCT+DISFCT(K)
330   CONTINUE
C
C CALCULATE BETA
```

```
      DO 340 K=1,5
      BETA(K)=DISFCT(K)/SUMFCT
340   CONTINUE
      DO 350 I=1,5
      IF(SPFLG(I).EQ.0) GO TO 350
      SATTST=BETA(I)*TOTCON
      IF(SATTST.LE.CHECK(I)) GO TO 350
      NPSLDG(J)=CHECK(I)*NPSFL(J)/1.E6
      WRITE(TTYOUT,360) J
350   CONTINUE
360   FORMAT(1X,'NON-POINT-SOURCE LOADING EXCEEDED SOLUBILITY LIMIT',
     1 ' IN ELEMENT ',I2,'.'/1X,'LOAD HAS BEEN ADJUSTED.')
      GO TO 390
370   CONTINUE
      NPSLDG(J)=0.
      WRITE(TTYOUT,380) J
380   FORMAT(1X,'A NON-POINT-SOURCE LOADING WAS ENTERED FOR ELEMENT ',
     1 I2,','/1X,'WITHOUT A NON-POINT SOURCE.  THE LOAD HAS BEEN ',
     2 'REMOVED.')
C
390   CONTINUE
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C COMPUTE TOTAL CORRECTED LOADS AND TOTAL SYSTEM LOAD:
      SYSLDL=0.
      DO 400 J=1,KOUNT
      TOTLDL(J)=IFLLDG(J)+STRLDG(J)+NPSLDG(J)+PCPLDG(J)+DRFLDG(J)
      SYSLDL=SYSLDL+TOTLDL(J)
400   CONTINUE
C IF NO NON-ZERO LOADS REMAIN, SYSTEM WILL NOT RUN:
      IF(SYSLDL.GT.0.) GO TO 420
      IFLAG=8
      WRITE(TTYOUT,410)
410   FORMAT(1X,'NO TOXICANT LOADINGS.  SYSTEM ABORTED.')
420   CONTINUE
      RETURN
      END
C FILE 'FIRORD.FTN'
C THIS SUBROUTINE COMBINES CHEMICAL AND ENVIRONMENTAL PARAMETERS TO
C A FIRST ORDER FORM FOR THE STEADY-STATE AND INTEGRATION ROUTINES.
      SUBROUTINE FIRORD
C-----------------------------------------------------------------------
C FIRORD. FIRORD REDUCES THE KINETICS OF THE TOXICANT TO PSEUDO-FIRST-
C ORDER FORM BY COUPLING CHARACTERISTICS OF THE TOXICANT AND THE
C ENVIRONMENT.  IF THE INPUT DATA PERMITS, THE PROPERTIES OF THE
C TOXICANT ARE ADJUSTED TO REFLECT THE EFFECT OF TEMPERATURE IN EACH
C PHYSICAL ELEMENT OF THE SYSTEM.  THE PSEUDO-FIRST-ORDER REACTION RATE
C CONSTANTS ARE COMPUTED FROM THE INTERACTIVE EFFECTS OF PARTITIONING,
C TEMPERATURE, AND OTHER CHARACTERISTICS OF THE ENVIRONMENT FOR EACH
C SPECIES OR FORM.  FIRORD ALSO CONVERTS THE INPUT LOADINGS AND
C ADVECTIVE AND DISPERSIVE FLOW FIELD TO PSEUDO-FIRST-ORDER EFFECTS ON
C TOXICANT CONCENTRATIONS, AND WRITES BOTH A KINETIC PROFILE OF THE
C TOXICANT AND A CANONICAL PROFILE OF THE ECOSYSTEM TO FILE
C 'REPORT.DAT'.  FIRORD ALSO CALLS 3 SUBROUTINES (PHOTO1, PHOTO2, AND
C VOLAT) THAT EVALUATE ASPECTS OF THE REACTION KINETICS.  (PHOTO1 AND
C PHOTO2 ARE MUTUALLY EXCLUSIVE COMPUTATIONS; FIRORD SELECTS THE
C APPROPRIATE ROUTINE BASED ON THE STRUCTURE OF THE INPUT DATA.)
```

```
C-----------------------------------------------------------------------ADL23950
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL23960
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL23970
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL23980
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL23990
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG     ADL24000
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL24010
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL24020
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL24030
     1,NPSEDG,INTFLG                                                    ADL24040
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH     ADL24050
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL24060
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL24070
      COMMON/NAME1G/CHEMNA(60)                                          ADL24080
      COMMON/CONTRG/SPFLG(5)                                            ADL24090
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL24100
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL24110
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL24120
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL24130
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL24140
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL24150
     1,KBHG(3,5),EBHG(3,5)                                              ADL24160
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL24170
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL24180
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL24190
     1,IFLLDG(10)                                                       ADL24200
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL24210
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL24220
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL24230
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL24240
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)     ADL24250
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)              ADL24260
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG         ADL24270
     1,DFACG(10)                                                        ADL24280
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                        ADL24290
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                       ADL24300
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)          ADL24310
     1,INTFLG(10)                                                       ADL24320
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL24330
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                      ADL24340
C---------------------------------------------------------------------- ADL24350
C FILE 'CHEML.COM'                                                      ADL24360
          COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)        ADL24370
          COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),    ADL24380
     2    VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                  ADL24390
          COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                     ADL24400
          COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                         ADL24410
          COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),         ADL24420
     2    CHEMPC,BIOPCT,VOLPCT,EXPPCT                                   ADL24430
          COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)             ADL24440
          REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                     ADL24450
C---------------------------------------------------------------------- ADL24460
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                   ADL24470
C---------------------------------------------------------------------- ADL24480
C FILE 'ENVIRL.COM'                                                     ADL24490
          COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                ADL24500
          COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),     ADL24510
     2  NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                           ADL24520
        COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,          ADL24530
     2  KDTIME,TFACTR,IPER,PERIOD                                       ADL24540
        COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)       ADL24550
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL24560
        COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),       ADL24570
     1     PSEDCO(10)                                                   ADL24580
        COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                ADL24590
     2 SEDFL(10,10),WATINL(10)                                          ADL24600
        COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),            ADL24610
     1  WAVEL(39),TOTETA(39),WLAML(39)                                  ADL24620
C**********                                                             ADL24630
      LOGICAL NPS,DOT                                                   ADL24640
      COMMON/NPSBLK/NPS,LUNPS,LUNPS2,NPSPER                             ADL24650
      COMMON/DOTBLK/DOT,LUDOT                                           ADL24660
C**********                                                             ADL24670
        REAL INTFL,NPSFL,NPSCOL                                         ADL24680
        REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                  ADL24690
        LOGICAL PRINT,PRINTO                                            ADL24700
        INTEGER L,E,H,B                                                 ADL24710
C-----------------------------------------------------------------------ADL24720
C-----------------------------------------------------------------------ADL24730
C LOCAL VARIABLES USED ONLY IN THIS SUBROUTINE:                         ADL24740
C ---------------------------------------------                         ADL24750
C EXPORTS FROM COMPARTMENTS THAT DO NOT LEAVE THE ECOSYSTEM:             ADL24760
        REAL INTOUL                                                     ADL24770
C KAH, KNH, KBH ARE HYDROLYSIS RATE CONSTANTS                            ADL24780
C KOX IS OXIDATION RATE CONSTANT                                        ADL24790
C KDP IS DIRECT PHOTOLYSIS RATE CONSTANT (/HR)                          ADL24800
C KBACWL IS BIOLYSIS RATE CONSTANT IN WATER COLUMN                      ADL24810
C KBACSL IS BIOLYSIS IN BOTTOM SEDIMENTS                                ADL24820
        REAL KAHL,KNHL,KBHL,KOXL,KDPL,KBACWL,KBACSL                     ADL24830
C ABSTOL IS THE TOTAL LIGHT ABSORBANCE OF THE 5 SPECIES - USED          ADL24840
C AS A CHECK ON DATA AVAILABILITY.                                      ADL24850
        DIMENSION ABSTOL(5)                                             ADL24860
C**********                                                             ADL24870
      PRINTO=PRINT                                                      ADL24880
      IF (DOT.OR.NPS) PRINT=.FALSE.                                     ADL24890
C**********                                                             ADL24900
C-----------------------------------------------------------------------ADL24910
C COMPUTE SUM OF ABSORBANCE DATA AS A CHECK ON DATA AVAILABILITY:       ADL24920
        DO 110 K=1,5                                                    ADL24930
        ABSTOL(K)=0.                                                    ADL24940
          DO 100 I=1,39                                                 ADL24950
          ABSTOL(K)=ABSTOL(K)+ABSG(I,K)                                 ADL24960
100       CONTINUE                                                      ADL24970
110     CONTINUE                                                        ADL24980
C-----------------------------------------------------------------------ADL24990
C WRITE COLUMN HEADINGS ETC. FOR TABLE OF PSEUDO-FIRST-ORDER KINETICS:  ADL25000
C TITLE FOR TABLE:                                                      ADL25010
        IF (PRINT) WRITE(RPTOUT,120)                                    ADL25020
120     FORMAT(1X,'KINETIC PROFILE OF ORGANIC TOXICANT.  RATE ',        ADL25030
     1'CONSTANTS DERIVED FROM'/1X,'COUPLING OF TOXICANT ',              ADL25040
     2'CHARACTERISTICS TO ECOSYSTEM PROPERTIES.')                       ADL25050
       IF (PRINT) WRITE(RPTOUT,130)                                     ADL25060
130     FORMAT(1X,72(1H-))                                              ADL25070
C COLUMN HEADINGS:                                                      ADL25080
```

```
      IF (PRINT) WRITE(RPTOUT,140)                                      ADL25090
140   FORMAT(1X,'CP T*',2X,11(1H-),1X,                                  ADL25100
     1'PSEUDO-FIRST-ORDER RATE CONSTANTS (/HR) ',                       ADL25110
     2 11(1H-))                                                         ADL25120
      IF (PRINT) WRITE(RPTOUT,150)                                      ADL25130
150   FORMAT(4X,'Y',3X,'HYDROLYSIS PHOTOLYSIS OXIDATION',               ADL25140
     1 2X,'BIOLYSIS',3X,'VOLATILITY',1X,'TRANSPORT'/1X,72(1H-))         ADL25150
C-----------------------------------------------------------------------ADL25160
C CALCULATE PSEUDO-FIRST-ORDER RATE CONSTANTS ETC. FOR                  ADL25170
C PRINTING KINETICS TABLE AND FOR COMPUTATIONS:                         ADL25180
C TAKE COMPARTMENTS IN NUMERICAL ORDER:                                 ADL25190
C-----------------------------------------------------------------------ADL25200
      DO 220 J=1,KOUNT                                                  ADL25210
C COMPUTE EFFECT OF *****-INPUT LOADINGS-***** ON CONCENTRATION:        ADL25220
      CONLDL(J)=1.E6*TOTLDL(J)/WATVOL(J)                                ADL25230
C COMPUTE FIRST ORDER RATE CONSTANTS FOR COMPARTMENT:                   ADL25240
C                                        ******                         ADL25250
C COMPUTE (SYSTEM AND INTERNAL TRANSPORT) EXPORT COEFFICIENTS:          ADL25260
C                                        ******                         ADL25270
C 1. EXPORTS FROM ECOSYSTEM:                                            ADL25280
      EXPOKL(J)=WATOUL(J)*ALPHA(16,J)+SEDOUL(J)*ALPHA(17,J)/            ADL25290
     1 SEDCOL(J)+WATOUL(J)*ALPHA(18,J)*PLRAG(J)                         ADL25300
C                                                                       ADL25310
C 2. EXPORTS FROM COMPARTMENTS THAT REMAIN IN ECOSYSTEM:                ADL25320
      INTOUL=0.                                                         ADL25330
      DO 160 I=1,KOUNT                                                  ADL25340
      INTOUL=INTOUL+WATFL(I,J)*ALPHA(16,J)+SEDFL(I,J)*ALPHA(17,J)/      ADL25350
     1 SEDCOL(J) + WATFL(I,J)*ALPHA(18,J)*PLRAG(J)                      ADL25360
160   CONTINUE                                                          ADL25370
C                                                                       ADL25380
C CONVERT TOTAL EXPORTS TO FIRST-ORDER EFFECT ON TOXICANT CONCENTRATION:ADL25390
      CONOUL=(EXPOKL(J)+INTOUL)/WATVOL(J)                               ADL25400
C   ------------------------------------------------------------------- ADL25410
C ZERO OUT PROCESS RATE CONSTANTS:                                      ADL25420
      HYDRKL(J)=0.                                                      ADL25430
      OXIDKL(J)=0.                                                      ADL25440
      PHOTKL(J)=0.                                                      ADL25450
      BIOLKL(J)=0.                                                      ADL25460
      VOLKL(J)=0.                                                       ADL25470
C        **************                                                 ADL25480
C COMPUTE VOLATILIZATION FROM COMPARTMENT:                              ADL25490
C        **************                                                 ADL25500
C ONLY THE DISSOLVED SH2 MOLECULE IN SURFACE WATERS CAN VOLATILIZE:     ADL25510
      IF(TYPEE(J).EQ.H .OR. TYPEE(J).EQ.B) GO TO 170                    ADL25520
C CHECK ON DATA AVAILABILITY:                                           ADL25530
      IF(MWTG.EQ.0. .OR. (HENRYG.EQ.0. .AND. VAPRG.EQ.0.)) GO TO 170    ADL25540
C CALL 2-FILM VOLATILIZATION SUBROUTINE, RETURN 1ST-ORDER RATE CONST.:  ADL25550
      CALL VOLAT(J,ALPHA(1,J),VOLKL(J))                                 ADL25560
170   CONTINUE                                                          ADL25570
C-----------------------------------------------------------------------ADL25580
C PRE-ZERO AVERAGE LIGHT INTENSITY IN COMPARTMENT (***-PHOTOLYSIS-***): ADL25590
      LIGHTL(J)=0.                                                      ADL25600
      KDPL=0.                                                           ADL25610
C COMPUTE ACTIVE BACTERIAL POPULATIONS FOR  ****-BIOLYSIS-***:          ADL25620
C   COMPUTE WATER COLUMN ACTIVE BACTERIA (CELLS/ML):                    ADL25630
      IF(INDEXW(J).EQ.1) ACBACL=BACTOG(J)*ACBACG(J)                     ADL25640
C   FOR SEDIMENTS,, CONVERT BACTERIAL POPULATIONS (CELLS/100 GRAMS      ADL25650
C   DRY WEIGHT OF SEDIMENT) TO ACTIVE CELLS/ML OF PORE WATER:           ADL25660
      IF(INDEXS(J).EQ.1) ACBACL=(BACTOG(J)*ACBACG(J)*SEDMSL(J))/       ADL25670
     1                             (WATVOL(J)*100.)                     ADL25680
C COMPUTE ACID/BASE CONCENTRATIONS FOR ***-HYDROLYSIS-***:              ADL25690
      HPLUS=10.**(-PHG(J))                                              ADL25700
      HYDRX=10.**(-POHG(J))                                             ADL25710
C KELVIN TEMPERATURE:                                                   ADL25720
      TKEL=TCELG(J)+273.15                                              ADL25730
C INITIALIZE COUNTER FOR COMPUTING INDEX FOR ALPHA VECTOR:              ADL25740
      II=-1                                                             ADL25750
C---------------------------------------------------------------------- ADL25760
C 5 SPECIES OF TOXICANT (SH2,SH3+,SH4++,SH-,SH=)                        ADL25770
      DO 210 K=1,5                                                      ADL25780
C SKIP OVER SPECIES THAT DO NOT OCCUR:                                  ADL25790
      IF(SPFLG(K).EQ.0) GO TO 200                                       ADL25800
C---------------------------------------------------------------------- ADL25810
C ***********                                                           ADL25820
C PHOTOLYSIS:                                                           ADL25830
C ***********                                                           ADL25840
C COMPUTE RATE CONSTANT FOR SPECIES, ADJUST BY REACTION QUANTUM YIELD   ADL25850
C AND DISTRIBUTION COEFFICIENT (ALPHA) IN NEXT (FORMS) LOOP:            ADL25860
C  CASE 1: NO PHOTOLYSIS IN SEDIMENT COMPARTMENTS:                      ADL25870
      IF (TYPEE(J).EQ.B) GO TO 180                                      ADL25880
C  CASE 2: DIRECT LOAD BY USER OF RATE CONSTANT @ 40 DEG. LATITUDE;     ADL25890
C          SUBROUTINE 'PHOTO1' WILL ADJUST FOR LATITUDE AND DEPTH       ADL25900
C          BASED ON COMPOSITE LIGHT EXTINCTION COEFFICIENT:             ADL25910
      IF(KDPG(K).GT.0.) CALL PHOTO1(KDPL,K,J)                           ADL25920
C  CASE 3: RATE CONSTANT NOT DIRECTLY LOADED, ABSORPTION SPECTRUM       ADL25930
C          AVAILABLE:                                                   ADL25940
      IF(KDPG(K).EQ.0. .AND. ABSTOL(K).GT.0.) CALL PHOTO2(KDPL,K,J)     ADL25950
180   CONTINUE                                                          ADL25960
C -----------------------------------------------------------------------ADL25970
C---------------------------------------------------------------------- ADL25980
C 3 FORMS OF EACH SPECIES (DISSOLVED, SORBED WITH SEDIMENT, BIOSORBED)  ADL25990
               DO 190 I=1,3                                             ADL26000
C CALCULATE ADDRESS FOR DISTRIBUTION COEFFICIENT (ALPHA):               ADL26010
      KK=2*K+II+I-1                                                     ADL26020
C                                                                       ADL26030
C       **********                                                      ADL26040
C ADJUST PHOTOLYSIS BY QUANTUM YIELD AND DISTRIBUTION:                  ADL26050
C       **********                                                      ADL26060
      PHOTKL(J)=PHOTKL(J)+ALPHA(KK,J)*KDPL*QUANTG(I,K)                  ADL26070
C --------------------------------------------------------------------- ADL26080
C **********                                                            ADL26090
C HYDROLYSIS:                                                           ADL26100
C ***********                                                           ADL26110
C LOAD THE GLOBAL (INPUT) VALUE OF THE RATE CONSTANTS AND ADJUST        ADL26120
C FOR ENVIRONMENTAL TEMPERATURE, ETC.:                                  ADL26130
C ACID HYDROLYSIS:                                                      ADL26140
               KAHL=KAHG(I,K)                                           ADL26150
               IF(EAHG(I,K).NE.0.) KAHL=10.**(KAHG(I,K)-((1000.*        ADL26160
     1          EAHG(I,K))/(4.58*TKEL)))                                ADL26170
C NEUTRAL HYDROLYSIS:                                                   ADL26180
               KNHL=KNHG(I,K)                                           ADL26190
               IF(ENHG(I,K).NE.0.) KNHL=10.**(KNHG(I,K)-((1000.*        ADL26200
     1          ENHG(I,K))/(4.58*TKEL)))                                ADL26210
C ALKALINE HYDROLYSIS:                                                  ADL26220
```

```
                KBHL=KBHG(I,K)
                IF(EBHG(I,K).NE.0.) KBHL=10.**(KBHG(I,K)-((1000.*
     1           EBHG(I,K))/(4.58*TKEL)))
C
C ADD CURRENT CONTRIBUTION TO TOTAL RATE CONSTANT:
        HYDRKL(J)=HYDRKL(J)+ALPHA(KK,J)*(KAHL*HPLUS+KNHL+KBHL*HYDRX)
C------------------------------------------------------------------------
C *********
C OXIDATION:
C *********
                KOXL=KOXG(I,K)
                IF(EOXG(I,K).NE.0.) KOXL=10.**(KOXG(I,K)-((1000.*
     1           EOXG(I,K))/(4.58*TKEL)))
C ADD CURRENT CONTRIBUTION TO TOTAL RATE CONSTANT:
        OXIDKL(J)=OXIDKL(J)+ALPHA(KK,J)*KOXL*OXRADG(J)
C -----------------------------------------------------------------------
C ********
C BIOLYSIS:
C ********
C LOAD GLOBAL (INPUT) VALUE OF RATE CONSTANTS:
                KBACWL=KBACWG(I,K)
                KBACSL=KBACSG(I,K)
                IF(QTBAWG(I,K).NE.0.) KBACWL=KBACWG(I,K)*
     1            QTBAWG(I,K)**((BIOTMG(J)-20.)/10.)
                IF(QTBASG(I,K).NE.0.) KBACSL=KBACSG(I,K)*
     1           QTBASG(I,K)**((BIOTMG(J)-20.)/10.)
C ADD CURRENT CONTRIBUTION TO TOTAL RATE CONSTANTS:
                BIOLKL(J)=BIOLKL(J)+ALPHA(KK,J)*ACBACL*
     1           (KBACWL*INDEXW(J) + KBACSL*INDEXS(J))
C-----------------------------------------------------------------------
190             CONTINUE
C END OF FORMS (DISSOLVED, SORBED, BIOSORBED) LOOP
C ----------------------------------------------------------------------
C INCREMENT ALPHA INDEX COUNTER:
200     II=II+1
210     CONTINUE
C END OF IONIC SPECIES LOOP
C COMPUTE TOTAL FIRST-ORDER RATE COEFFICIENT:
C------------------------------------------------------------------------
        TOTKL(J)=CONOUL+VOLKL(J)+PHOTKL(J)+BIOLKL(J)+OXIDKL(J)+HYDRKL(J)
C WRITE LINE TO REPORT:
      IF (PRINT)
     $  WRITE(RPTOUT,230) J,TYPEE(J),HYDRKL(J),PHOTKL(J),OXIDKL(J),
     1  BIOLKL(J),VOLKL(J),CONOUL
220     CONTINUE
C END OF COMPARTMENT LOOP
C----------------------------------------------------------------------
230     FORMAT(1X,I2,1X,1A1,1X,6(1X,1PG10.3))
C----------------------------------------------------------------------
C PRINT FOOTNOTE TO TABLE:
        IF (PRINT) WRITE(RPTOUT,240)
240   FORMAT(1X,72(1H-)/1X,'* COMP. TYPE: "L"=LITTORAL; "E"=(EPI) AND',
     1  ' "H"=(HYPO)LIMNION; "B"=BENTHIC')
C-------------------------------------*************************------
C COMPUTE FIRST ORDER COEFFICIENTS FOR INTERNAL TRANPORT INPUTS:
C (AND TANSPOSE MATRIX) ----------------*************************------
        DO 260 J=1,KOUNT
```

```
         DO 250 I=1,KOUNT                                                 ADL26800
          INTINL(I,J)=WATFL(J,I)*ALPHA(16,I)+SEDFL(J,I)*ALPHA(17,I)/      ADL26810
     1     SEDCOL(I) + WATFL(J,I)*ALPHA(18,I)*PLRAG(I)                    ADL26820
C CONVERT INPUT TO FACTOR EXPRESSING CONCENTRATION CHANGE IN RECEIVING    ADL26830
C COMPARTMENT:                                                            ADL26840
           INTINL(I,J)=INTINL(I,J)/WATVOL(J)                              ADL26850
250       CONTINUE                                                        ADL26860
260      CONTINUE                                                         ADL26870
C-----------------------------------------------------------------------  ADL26880
C PRINT CANONICAL PROFILE OF ENVIRONMENT IN REPORT:                       ADL26890
C PAGE HEADER:                                                            ADL26900
         IF (PRINT) WRITE(RPTOUT,270) CHEMNA,SYSTYP                       ADL26910
270      FORMAT(1H1,'  AERL-ESB MODEL OF FATE OF ORGANIC TOXICANTS',      ADL26920
     11X,'IN AQUATIC ECOSYSTEMS.'/1X,'CHEMICAL: ',60A1/1X,                ADL26930
     2'ECOSYSTEM: ',60A1/1X,72(1H-))                                      ADL26940
C SET UP TABLE FOR REPORT:                                                ADL26950
         IF (PRINT) WRITE(RPTOUT,280)                                     ADL26960
280      FORMAT(21X,'CANONICAL PROFILE OF ECOSYSTEM'/1X,72(1H-))          ADL26970
C COLUMN HEADINGS:                                                        ADL26980
         IF (PRINT) WRITE(RPTOUT,290)                                     ADL26990
290      FORMAT(1X,'CP T*  PH   POH  TEMP REAERATION COMPOSITE ',         ADL27000
     1   'BACTERIAL  OXIDANT  DISSOLVED'/4X,'Y',13X,'DEG.   COEFF.',3X,   ADL27010
     2   'LIGHT AVE POP. SIZE   CONC.    PERCENT'/4X,'P',14X,'C.',        ADL27020
     3   4X,'M/HR',9X,'%',5X,'CELLS/**',3X,'(MOLAR)',6X,'%'/1X,72(1H-))   ADL27030
         DO 300 J=1,KOUNT                                                 ADL27040
C CONVERT LIGHT INDEX TO PERCENTAGE:                                      ADL27050
         TEMP1=LIGHTL(J)*100.                                             ADL27060
         TEMP2=ACBACG(J)*BACTOG(J)                                        ADL27070
C CONVERT FRACTION OF TOXICANT PRESENT IN DISSOLVED FORMS TO PERCENT:     ADL27080
         TEMP3=ALPHA(16,J)*100.                                           ADL27090
      IF (PRINT)                                                          ADL27100
     $   WRITE(RPTOUT,310) J,TYPEE(J),PHG(J),POHG(J),TCELG(J),KO2L(J),    ADL27110
     1   TEMP1,TEMP2,OXRADG(J),TEMP3                                      ADL27120
300      CONTINUE                                                         ADL27130
C FORMAT FOR OUTPUT:                                                      ADL27140
310      FORMAT(1X,I2,1X,A1,2F6.2,F5.1,1X,1PG10.3,4G10.3)                 ADL27150
C WRITE FOOTNOTE TO TABLE:                                                ADL27160
         IF (PRINT) WRITE(RPTOUT,320)                                     ADL27170
320    FORMAT(1X,72(1H-)/1X,'* COMP. TYPE: "L"=LITTORAL; "E"=(EPI) AND',  ADL27180
     1  ' "H"=(HYPO)LIMNION; "B"=BENTHIC'/1X,'** ACTIVE BACTERIAL ',      ADL27190
     2 ' POPULATIONS AS CELLS/ML IN WATER COLUMN,'/4X,                    ADL27200
     3 'CELLS/100 G (DRY WEIGHT) OF SEDIMENTS IN BOTTOM SEDIMENTS.')      ADL27210
C-----------------------------------------------------------------------  ADL27220
C-----------------------------------------------------------------------  ADL27230
C**********                                                              ADL27240
      PRINT=PRINTO                                                        ADL27250
C**********                                                              ADL27260
        RETURN                                                            ADL27270
      END                                                                 ADL27280
C FILE 'PHOTO1.FTN'                                                       ADL27290
        SUBROUTINE PHOTO1(RATEK,K,J)                                      ADL27300
C------------------------------------------------------------------------ADL27310
C PHOTO1. PHOTO1 ACCEPTS A MEASURED (CLEAR DAY) PHOTOLYSIS RATE CONSTANTADL27320
C AT A SPECIFIED REFERENCE LATITUDE AS INPUT DATA.  PHOTO1 CORRECTS THISADL27330
C RATE CONSTANT FOR DEVIATION OF THE LATITUDE OF THE ECOSYSTEM FROM THE ADL27340
C REFERENCE LATITUDE, EFFECTS OF CLOUD COVER, AND LIGHT EXTINCTION IN   ADL27350
C THE WATER COLUMN BASED ON A COMPOSITE ZENITH LIGHT EXTINCTION         ADL27360
```

```
C COEFFICIENT ENTERED AS PART OF THE CANONICAL ENVIRONMENTAL DATA.      ADL27370
C-----------------------------------------------------------------------ADL27380
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL27390
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL27400
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL27410
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL27420
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG     ADL27430
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL27440
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL27450
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL27460
     1,NPSEDG,INTFLG                                                    ADL27470
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH     ADL27480
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL27490
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL27500
      COMMON/NAME1G/CHEMNA(60)                                          ADL27510
      COMMON/CONTRG/SPFLG(5)                                            ADL27520
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL27530
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL27540
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL27550
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL27560
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL27570
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL27580
     1,KBHG(3,5),EBHG(3,5)                                              ADL27590
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL27600
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL27610
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL27620
     1,IFLLDG(10)                                                       ADL27630
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL27640
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL27650
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL27660
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL27670
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)     ADL27680
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)              ADL27690
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG         ADL27700
     1,DFACG(10)                                                        ADL27710
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                        ADL27720
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                       ADL27730
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)          ADL27740
     1,INTFLG(10)                                                       ADL27750
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL27760
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                      ADL27770
C---------------------------------------------------------------------  ADL27780
C FILE 'CHEML.COM'                                                      ADL27790
        COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)          ADL27800
        COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),      ADL27810
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                    ADL27820
        COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                       ADL27830
        COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                           ADL27840
        COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),           ADL27850
     2  CHEMPC,BIOPCT,VOLPCT,EXPPCT                                     ADL27860
        COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)               ADL27870
        REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                       ADL27880
C---------------------------------------------------------------------  ADL27890
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                   ADL27900
C---------------------------------------------------------------------  ADL27910
C FILE 'ENVIRL.COM'                                                     ADL27920
        COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                  ADL27930
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),         ADL27940
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                            ADL27950
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,            ADL27960
     2 KDTIME,TFACTR,IPER,PERIOD                                        ADL27970
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)         ADL27980
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL27990
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),         ADL28000
     1    PSEDCO(10)                                                    ADL28010
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                  ADL28020
     2 SEDFL(10,10),WATINL(10)                                          ADL28030
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),              ADL28040
     1 WAVEL(39),TOTETA(39),WLAML(39)                                   ADL28050
      REAL INTFL,NPSFL,NPSCOL                                           ADL28060
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                    ADL28070
      LOGICAL PRINT                                                     ADL28080
      INTEGER L,E,H,B                                                   ADL28090
C---------------------------------------------------------------------- ADL28100
C-----------------------------------------------------------------------ADL28110
      REAL IZERO                                                        ADL28120
C-----------------------------------------------------------------------ADL28130
C LIGHT EXTINCTION CORRECTION FACTOR IS COMPUTED AT THE TIME OF THE     ADL28140
C FIRST CALL TO THIS SUBROUTINE FOR EACH COMPARTMENT.  IF THE CURRENT   ADL28150
C CALL IS INVOKED BY AN ADDITIONAL CHEMICAL SPECIES, THE LIGHT          ADL28160
C COMPUTATIONS ARE SKIPPED.                                             ADL28170
      IF (J.EQ.JSAV1) GO TO 200                                         ADL28180
      JSAV1=J                                                           ADL28190
C-----------------------------------------------------------------------ADL28200
C THE DISTRIBUTION FACTOR (DFACG) IS THE OPTICAL PATH IN THE COMPARTMENTADL28210
C AS A RATIO TO THE DEPTH.  IT IS DIFFICULT TO COMPUTE, BUT A PROBABLE  ADL28220
C BEST VALUE IS 1.19 (HUTCHINSON, TREATISE LIMNOLOGY).  HOWEVER, IN     ADL28230
C THE PRESENCE OF A LARGE CONCENTRATION OF SCATTERING PARTICLES, IT     ADL28240
C MAY APPROACH OR REACH 2.0.  IN ORDER TO ENSURE THAT AN IMPROPER VALUE ADL28250
C IS NOT LOADED AND USED IN COMPUTATIONS, THE INPUT DFACG IS CHECKED    ADL28260
C AND SET TO 1.19 IF THE INPUT IS INVALID.                              ADL28270
      IF (DFACG(J).LT.1. .OR. DFACG(J).GT.2.) DFACG(J)=1.19             ADL28280
C COMPUTATION IS SKIPPED SINCE LIGHTL(J) ALREADY CONTAINS THE LIGHT     ADL28290
C COMPOSITE LIGHT FIELD:                                                ADL28300
C COMPUTE AVERAGE LIGHT LEVEL (AS FRACTION OF INCIDENT LIGHT) FROM      ADL28310
C THE COMPOSITE ZENITH LIGHT EXTINCTION COEFFICIENT ENTERED BY USER:    ADL28320
C                                                                       ADL28330
C COMPARTMENT 1 IS NECESSARILY A WATER COLUMN WITHOUT AN OVERLYING      ADL28340
C WATER MASS:                                                           ADL28350
      IF(J-1) 160,150,160                                               ADL28360
150   CONTINUE                                                          ADL28370
C                                                                       ADL28380
C COMPUTE LIGHT LEVEL AT BOTTOM OF COMPARTMENT:                         ADL28390
      BOTLIT=EXP(-(CMPETG(1)*DEPTHG(1)*DFACG(J)))                       ADL28400
      LIGHTL(1)=(1.-BOTLIT)/(CMPETG(1)*DEPTHG(1)*DFACG(J))              ADL28410
      GO TO 200                                                         ADL28420
C                                                                       ADL28430
C                                                                       ADL28440
160   CONTINUE                                                          ADL28450
C THIS COMPARTMENT IS A WATER COLUMN NUMBERED GREATER THAN 1, BUT       ADL28460
C THE COMPUTATION DEPENDS ON WHETHER THERE IS ANOTHER EPILIMNION        ADL28470
C OR HYPOLIMNION COMPARTMENT ABOVE IT:                                  ADL28480
      IF(TYPEE(J-1).EQ.E .OR. TYPEE(J-1).EQ.H) GO TO 180                ADL28490
C                                                                       ADL28500
```

```
C NO OVERLYING WATER MASS: SURFACE LIGHT INTENSITY (IZERO) IS ONE:      ADL28510
        BOTLIT=EXP(-(CMPETG(J)*DEPTHG(J)*DFACG(J)))                     ADL28520
        LIGHTL(J)=(1.-BOTLIT)/(CMPETG(J)*DEPTHG(J)*DFACG(J))            ADL28530
        GO TO 200                                                       ADL28540
C                                                                       ADL28550
C                                                                       ADL28560
180     CONTINUE                                                        ADL28570
C WATER COLUMN WITH AN OVERLYING WATER MASS: BOTLIT CONTAINS THE        ADL28580
C (RELATIVE) LIGHT INTENSITY AT THE TOP OF THE CURRENT COMPARTMENT:     ADL28590
        IZERO=BOTLIT                                                    ADL28600
        BOTLIT=IZERO*EXP(-(CMPETG(J)*DEPTHG(J)*DFACG(J)))               ADL28610
        LIGHTL(J)=(IZERO-BOTLIT)/(CMPETG(J)*DEPTHG(J)*DFACG(J))         ADL28620
C                                                                       ADL28630
C                                                                       ADL28640
200     CONTINUE                                                        ADL28650
C                                                                       ADL28660
C-----------------------------------------------------------------------ADL28670
C COMPUTE LATITUDE CORRECTION FACTOR:                                   ADL28680
        FACTOR=(191696.65+87054.63*COS(0.0349*LATG))/(191696.65+        ADL28690
     1  87054.63*COS(0.0349*RFLATG(K)))                                 ADL28700
C-----------------------------------------------------------------------ADL28710
C ADJUST RATE FOR CLOUDINESS, LATITUDE, LIGHT EXTINCTION:               ADL28720
        RATEK=KDPG(K)*LIGHTL(J)*(1.-0.056*CLOUDG)*FACTOR                ADL28730
C-----------------------------------------------------------------------ADL28740
        RETURN                                                          ADL28750
        END                                                             ADL28760
C FILE 'PHOTO2.FTN'                                                     ADL28770
C SUBROUTINE TO COMPUTE PHOTOLYSIS RATE CONSTANTS FROM ABSORPTION       ADL28780
C SPECTRUM OF TOXICANT AND PROPERTIES OF THE ECOSYSTEM.                 ADL28790
C-----------------------------------------------------------------------ADL28800
C PHOTO2. PHOTO2 COMPUTES THE PHOTOLYSIS RATE CONSTANT BY COUPLING THE  ADL28810
C ABSORPTION SPECTRUM OF THE TOXICANT TO THE AMBIENT LIGHT FIELD.  LIGHTADL28820
C INTENSITY IN THE WATER COLUMN IS COMPUTED BY ASSEMBLING A TOTAL       ADL28830
C DIFFUSE ATTENUATION COEFFICIENT FROM THE INPUT DATA (DOC, CHLOROPHYLL,ADL28840
C SUSPENDED SEDIMENT CONCENTRATIONS), AND COUPLING THIS INFORMATION TO  ADL28850
C AN INPUT-SPECIFIED NEAR-SURFACE SPECTRAL IRRADIANCE AND DISTRIBUTION  ADL28860
C COEFFICIENT.                                                          ADL28870
C-----------------------------------------------------------------------ADL28880
         SUBROUTINE PHOTO2(RATEK,K,J)                                   ADL28890
C-----------------------------------------------------------------------ADL28900
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL28910
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL28920
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL28930
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL28940
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG     ADL28950
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL28960
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL28970
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL28980
     1,NPSEDG,INTFLG                                                    ADL28990
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH     ADL29000
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL29010
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL29020
      COMMON/NAME1G/CHEMNA(60)                                          ADL29030
      COMMON/CONTRG/SPFLG(5)                                            ADL29040
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL29050
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL29060
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL29070
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL29080
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL29090
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL29100
     1,KBHG(3,5),EBHG(3,5)                                              ADL29110
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL29120
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL29130
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL29140
     1,IFLLDG(10)                                                       ADL29150
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL29160
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL29170
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL29180
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL29190
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)     ADL29200
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)              ADL29210
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG         ADL29220
     1,DFACG(10)                                                        ADL29230
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                        ADL29240
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                       ADL29250
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)          ADL29260
     1,INTFLG(10)                                                       ADL29270
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL29280
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                      ADL29290
C---------------------------------------------------------------------- ADL29300
C FILE 'CHEML.COM'                                                      ADL29310
        COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)          ADL29320
        COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),      ADL29330
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                    ADL29340
        COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                       ADL29350
        COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                           ADL29360
        COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),           ADL29370
     2  CHEMPC,BIOPCT,VOLPCT,EXPPCT                                     ADL29380
        COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)               ADL29390
        REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                       ADL29400
C---------------------------------------------------------------------- ADL29410
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                   ADL29420
C---------------------------------------------------------------------- ADL29430
C FILE 'ENVIRL.COM'                                                     ADL29440
        COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                  ADL29450
        COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),       ADL29460
     2  NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                           ADL29470
        COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,          ADL29480
     2  KDTIME,TFACTR,IPER,PERIOD                                       ADL29490
        COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)       ADL29500
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL29510
        COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),       ADL29520
     1     PSEDCO(10)                                                   ADL29530
        COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                ADL29540
     2 SEDFL(10,10),WATINL(10)                                          ADL29550
        COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),            ADL29560
     1  WAVEL(39),TOTETA(39),WLAML(39)                                  ADL29570
        REAL INTFL,NPSFL,NPSCOL                                         ADL29580
        REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                  ADL29590
        LOGICAL PRINT                                                   ADL29600
        INTEGER L,E,H,B                                                 ADL29610
C---------------------------------------------------------------------- ADL29620
C-----------------------------------------------------------------------ADL29630
C K IS THE NUMBER OF THE CHEMICAL SPECIES (1=SH2, 2=SH3+, 3=SH4++,      ADL29640
C 4=SH-, 5=SH=), J IS THE NUMBER OF THE ECOSYSTEM COMPARTMENT BEING     ADL29650
C PROCESSED IN 'FIRORD' THAT INVOKED THIS CALL TO PHOTO2.               ADL29660
C ENTRY TO THIS SUBROUTINE IS INVOKED WHEN A DIRECT PHOTOLYSIS RATE      ADL29670
C CONSTANT HAS NOT BEEN LOADED IN THE CHEMICAL DATA, BUT THE ABSORPTION ADL29680
C SPECTRUM OF THE COMPOUND  IS AVAILABLE.  THE COMPARTMENT TO BE        ADL29690
C PROCESSED IS NECESSARILY A WATER COLUMN, SINCE THE SUBROUTINE IS NOT  ADL29700
C INVOKED FOR SEDIMENT COMPARTMENTS.  THE CHEMICAL SPECIES ARE          ADL29710
C IDENTIFIED, AND THE SPECTRA ARE LOCATED, FROM THE PARAMETER K PASSED  ADL29720
C IN FROM SUBROUTINE 'FIRORD'.                                          ADL29730
C-----------------------------------------------------------------------ADL29740
C LOCAL VARIABLES FOR THIS SUBROUTINE:  FACTOR, LAMBDA, SEDFAC, TOTAVE, ADL29750
C TOTTOP, AND WTEMP.                                                     ADL29760
      DIMENSION WTEMP(39)                                                ADL29770
C---------------------------------------------------------------------- ADL29780
C IF THIS IS THE INITIAL CALL TO PHOTO2, THE SURFACE (INPUT) LIGHT      ADL29790
C FIELD IS ADJUSTED FOR EFFECTS OF CLOUDINESS AND FOR THE WAVELENGTH    ADL29800
C INTERVALS (E.G. FOR 750 NM, THE INPUT DATA IS PER 10 NM, BUT THE      ADL29810
C TOTAL WAVELENGTH INTERVAL IN 50 NM, HENCE THE 5* CORRECTION).         ADL29820
C THESE COMPUTATIONS ARE PERFORMED ONCE, AND CAN BE SKIPPED AFTER       ADL29830
C THE INITIAL CALL TO PHOTO2.                                            ADL29840
      IF (ICALL.EQ.1) GO TO 70                                           ADL29850
      ICALL=1                                                            ADL29860
C ALL THE COMPUTATIONS INCORPORATE A LINEAR NUMERICAL CONVERSION        ADL29870
C FACTOR WITH SEVERAL COMPONENTS; THIS FACTOR IS PRE-COMPUTED AT THIS   ADL29880
C POINT AND THE LOCAL VALUES FOR OVERALL LIGHT INTENSITY ARE PRE-       ADL29890
C MULTIPLIED BY THE FACTOR.                                              ADL29900
      FACTOR = 2.3026*1000.*3600.*(1-0.056*CLOUDG)/6.023E23              ADL29910
C              LN 10. ML/L  SEC/HR                 AVOGADRO'S #          ADL29920
C NATURAL LOG OF 10. = 2.3026 (CONVERSION OF DECADIC MOLAR EXTINCTION   ADL29930
C COEFFICIENTS FOR TOXICANT TO NAPERIAN BASIS).                         ADL29940
C FACTOR OF 1000. CORRECTS VOLUME UNITS (LIGHT IS /SQ.CM., TOXICANT     ADL29950
C ABSORPTION IS /CM, OUTCOME IS ON A CC. VOLUMETRIC BASIS, MULTIPLIED    ADL29960
C BY 1000. CC/LITER).                                                    ADL29970
C DIVISION BY AVOGADRO'S NUMBER (6.023E23) CONVERTS PHOTONS TO MOLAR    ADL29980
C BASIS.                                                                 ADL29990
      DO 10 LAMBDA=1,28                                                  ADL30000
      WLAML(LAMBDA)=WLAMG(LAMBDA)*FACTOR                                 ADL30010
10    CONTINUE                                                           ADL30020
      WLAML(29)=WLAMG(29)*1.75*FACTOR                                    ADL30030
      DO 20 LAMBDA=30,36                                                 ADL30040
      WLAML(LAMBDA)=WLAMG(LAMBDA)*2.5*FACTOR                             ADL30050
20    CONTINUE                                                           ADL30060
      WLAML(37)=WLAMG(37)*3.75*FACTOR                                    ADL30070
      WLAML(38)=WLAMG(38)*5.0*FACTOR                                     ADL30080
      WLAML(39)=WLAMG(39)*5.0*FACTOR                                     ADL30090
70    CONTINUE                                                           ADL30100
C-----------------------------------------------------------------------ADL30110
C-----------------------------------------------------------------------ADL30120
C NEXT COMPUTATIONS ARE FOR THE CURRENT COMPARTMENT, AND CAN BE SKIPPED ADL30130
C IF THIS IS A CALL TO PHOTO2 FOR EVALUATION OF AN ADDITIONAL CHEMICAL  ADL30140
C SPECIES (E.G. AN ANION OR CATION EVALUATED AFTER THE NEUTRAL MOLECULE ADL30150
C HAS BEEN COMPUTED).                                                    ADL30160
      IF(J.EQ.JSAV2) GO TO 500                                           ADL30170
      JSAV2=J                                                            ADL30180
C THE DISTRIBUTION FACTOR DFACG(J) IS THE RELATIVE OPTICAL PATH IN      ADL30190
C THE SYSTEM.  THE BEST VALUE TO USE IS ABOUT 1.19 (HUTCHINSON), BUT    ADL30200
C IT INCREASES WITH DEPTH AND DUE TO THE PRESENCE OF LIGHT-SCATTERING   ADL30210
```

```
C PARTICLES.  ALTHOUGH IT IS AVAILABLE TO THE USER AS A PIECE OF INPUT
C DATA IT IS INTERNALLY CONSTRAINED TO VALUES BETWEEN 1. AND 2. SINCE
C THIS IS THE ACTUAL POSSIBLE RANGE OF THE PARAMETER.
        IF (DFACG(J).LT.1. .OR. DFACG(J).GT.2.) DFACG(J)=1.19
C
C THE VALUE FOR THE SPECIFIC DIFFUSE ATTENUATION COEFFICIENT FOR
C SUSPENDED SEDIMENTS (0.54 /M /(MG/L) ) IS FROM MILLER & ZEPP 1979
C (WATER RESEARCH).
        SEDFAC=0.54*SDCHRG(J)
C COMPUTE TOTAL DIFFUSE ATTENUATION COEFFICIENTS:
        DO 40 LAMBDA=1,39
        TOTETA(LAMBDA)=WATETA(LAMBDA)+PIGETA(LAMBDA)*CHLG(J)+
     1                 DOCETA(LAMBDA)*DOCG(J)+SEDFAC
40      CONTINUE
C COMPUTE AVERAGE AND BOTTOM LIGHT INTENSITIES FOR COMPARTMENT
C COMPARTMENT ONE IS NECESSARILY A WATER COLUMN WITH NO OVERLYING
C WATER MASS:
        IF (J.EQ.1) GO TO 100
C DOES THE CURRENT COMPARTMENT (J.GT.1) HAVE AN OVERLYING WATER MASS:
        IF (TYPEE(J-1).EQ.E .OR. TYPEE(J-1).EQ.H) GO TO 120
100     CONTINUE
C COMPARTMENT DOES NOT HAVE AN OVERLYING WATER MASS, LOAD LIGHT
C LEVEL AT TOP OF COMPARTMENT:
        DO 110 LAMBDA=1,39
          WTEMP(LAMBDA)=WLAML(LAMBDA)
110     CONTINUE
        GO TO 140
C
120     CONTINUE
C COMPARTMENT HAS AN OVERLYING WATER MASS, BOTLAM CONTAINS THE
C LIGHT INTENSITY AT THE BOTTOM OF THE PREVIOUS COMPARTMENT:
        DO 130 LAMBDA=1,39
          WTEMP(LAMBDA)=BOTLAM(LAMBDA)
130     CONTINUE
140     CONTINUE
C ZERO SUMS USED TO REPORT AVERAGE IRRADIANCE TO USER:
        TOTTOP=0.
        TOTAVE=0.
C COMPUTE LIGHT AT BOTTOM OF COMPARTMENT AND AVERAGE LIGHT LEVEL:
        DO 150 LAMBDA=1,39
          BOTLAM(LAMBDA)=WTEMP(LAMBDA)*EXP(-(TOTETA(LAMBDA)*DEPTHG(J)))
          IF(ABSG(LAMBDA,K).GT.0.) TOTTOP=TOTTOP+WLAML(LAMBDA)
          WAVEL(LAMBDA)=(WTEMP(LAMBDA)-BOTLAM(LAMBDA))/
     1                  (TOTETA(LAMBDA)*DEPTHG(J))
          IF(ABSG(LAMBDA,K).GT.0.) TOTAVE=TOTAVE+WAVEL(LAMBDA)
150     CONTINUE
C
        IF (TOTTOP.GT.0.) LIGHTL(J)=TOTAVE/TOTTOP
C-----------------------------------------------------------------------
500     CONTINUE
C COMPUTE LIGHT ABSORPTION BY TOXICANT:
        RATEK=0
        DO 510 LAMBDA=1,39
        RATEK=RATEK+DFACG(J)*WAVEL(LAMBDA)*ABSG(LAMBDA,K)
510     CONTINUE
C-----------------------------------------------------------------------
C
```

```
C
      RETURN
      END
C FILE 'VOLAT.FTN'
      SUBROUTINE VOLAT(K,BETA,RATEK)
C-----------------------------------------------------------------------
C VOLAT. VOLAT COMPUTES VOLATILIZATION RATE CONSTANTS USING THE TWO-FILM
C MODEL OF MOVEMENT OF TOXICANT ACROSS THE AIR-WATER INTERFACE.
C-----------------------------------------------------------------------
C BETA IS THE FRACTION OF THE TOXICANT PRESENT AS THE NEUTRAL DISSOLVED
C MOLECULE, RATEK IS THE FIRST-ORDER RATE COEFFICIENT TO BE COMPUTED.
C K IS THE NUMBER OF THE CURRENT COMPARTMENT.
C----------------------------------------------------------------------
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG
     1,NPSEDG,INTFLG
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT
      LOGICAL*1 CHEMNA,SYSTYP
      COMMON/NAME1G/CHEMNA(60)
      COMMON/CONTRG/SPFLG(5)
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)
     1,KBHG(3,5),EBHG(3,5)
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)
     1,IFLLDG(10)
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG
     1,DFACG(10)
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)
     1,INTFLG(10)
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:
C----------------------------------------------------------------------
C FILE 'CHEML.COM'
         COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)
         COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),
     2   VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)
```

```
      COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)
      COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV
      COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),
     2 CHEMPC,BIOPCT,VOLPCT,EXPPCT
      COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L
C-----------------------------------------------------------------------
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:
C-----------------------------------------------------------------------
C FILE 'ENVIRL.COM'
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,
     2 KDTIME,TFACTR,IPER,PERIOD
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)
     1,JSAV1,JSAV2,ICALL,NPX,NCON
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),
     1    PSEDCO(10)
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),
     2 SEDFL(10,10),WATINL(10)
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),
     1 WAVEL(39),TOTETA(39),WLAML(39)
      REAL INTFL,NPSFL,NPSCOL
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL
      LOGICAL PRINT
      INTEGER L,E,H,B
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C EVALUATE OXYGEN TRANSFER RATE FROM INPUT DATA:
C REAERATION IN CM/HR @ 20 DEG. C. IN INPUT DATA, HERE CONVERTED TO
C UNITS OF METERS/HOUR AND ADJUSTED FOR LOCAL TEMPERATURE
      KO2L(K)=(KO2G(K)/100.)*(1.024**(TCELG(K)-20.))
C-----------------------------------------------------------------------
      TKEL = 273.15 + TCELG(K)
C-----------------------------------------------------------------------
C LIQUID PHASE RESISTANCE: USE EXPERIMENTAL VALUE IF AVAILABLE,
C OTHERWISE ESTIMATE FROM MOLECULAR WEIGHT:
      IF(KVOG.GT.0.) GO TO 100
C COMPUTE LIQUID PHASE RESISTANCE:
      RESLIQ=1./(KO2L(K)*SQRT(32./MWTG))
      GO TO 150
100   CONTINUE
      RESLIQ=1./(KO2L(K)*KVOG)
C
150   CONTINUE
C-----------------------------------------------------------------------
C COMPUTE GAS PHASE RESISTANCE:
C ----------------------------
      IF(HENRYG.GT.0.) GO TO 500
C
C COMPUTE HENRY'S LAW CONSTANT FROM VAPOR PRESSURE AND SOLUBILITY:
C CONVERT VAPOR PRESSURE TO ATMOSPHERES:
      VAPRL=VAPRG/760.
C
C TEMPERATURE CORRECTION:
      IF(EVPRG.NE.0.) VAPRL=(10.**(VAPRG-((1000.*EVPRG)/(4.58*TKEL)
     2  )))/760.                                                         ADL31930
C                                                                        ADL31940
C COMPUTE SOLUBILITY AS MOLES PER CUBIC METER:                           ADL31950
C                                                                        ADL31960
       SOLL=SOLG(1)                                                      ADL31970
C TEMPERATURE CORRECTION:                                                ADL31980
       IF(ESOLG(1).NE.0.) SOLL=(10.**(SOLG(1)-((1000.*ESOLG(1))/         ADL31990
     2  (4.58*TKEL))))*MWTG*1000.                                        ADL32000
C                                                                        ADL32010
C CONVERT MG/L TO MOLES/CU. M.                                           ADL32020
       SOLL=SOLL/MWTG                                                    ADL32030
C                                                                        ADL32040
       HENRYL=VAPRL/SOLL                                                 ADL32050
C                                                                        ADL32060
       GO TO 600                                                         ADL32070
500    CONTINUE                                                          ADL32080
       HENRYL=HENRYG                                                     ADL32090
600    CONTINUE                                                          ADL32100
C                                                                        ADL32110
C COMPUTE WATER VAPOR TRANSPORT TERM A FUNCTION OF WINDSPEED (LISS '73)  ADL32120
       WAT=.1857 + 11.36 * WINDG(K)                                      ADL32130
C                                                                        ADL32140
C GAS PHASE RESISTANCE:                                                  ADL32150
       RESGAS = 1./ ((WAT*HENRYL*SQRT(18./MWTG))/(TKEL*8.206E-5))        ADL32160
C   (GAS CONSTANT IS 8.206E-5, MOL WT OF WATER IS 18.)                   ADL32170
C----------------------------------------------------------------------  ADL32180
C COMPUTE OVERALL FIRST ORDER VOLATILIZATION COEFFICIENT:                ADL32190
       COND=1./(RESLIQ+RESGAS)                                           ADL32200
       RATEK=(COND*BETA)/DEPTHG(K)                                       ADL32210
C----------------------------------------------------------------------  ADL32220
       RETURN                                                            ADL32230
       END                                                               ADL32240
C FILE 'STEADY.FTN'                                                      ADL32250
       SUBROUTINE STEADY(Y)                                              ADL32260
C----------------------------------------------------------------------  ADL32270
C STEADY. STEADY COMPUTES THE STEADY-STATE CONCENTRATIONS OF TOXICANT    ADL32280
C IN ALL SYSTEM ELEMENTS (USING AN ITERATIVE FIRST-ORDER CASCADE         ADL32290
C TECHNIQUE) INCLUDING THE EFFECTS OF INTERNAL TRANSPORTS AMONG          ADL32300
C COMPARTMENTS.  IF SOLUBILITY CRITERIA ARE VIOLATED AS A RESULT OF      ADL32310
C DRIFT LOADINGS, THESE LOADS ARE REDUCED AND THE COMPUTATIONS ARE       ADL32320
C RE-EVALUATED.  STEADY THEN WRITES A FINAL PROFILE OF THE LOADS TO FILEADL32330
C 'REPORT.DAT'.                                                          ADL32340
C-----------------------------------------------------------------------ADL32350
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:      ADL32360
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG     ADL32370
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG  ADL32380
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG    ADL32390
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG      ADL32400
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG  ADL32410
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG     ADL32420
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG     ADL32430
     1,NPSEDG,INTFLG                                                     ADL32440
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH      ADL32450
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                           ADL32460
      LOGICAL*1 CHEMNA,SYSTYP                                            ADL32470
      COMMON/NAME1G/CHEMNA(60)                                           ADL32480
      COMMON/CONTRG/SPFLG(5)                                             ADL32490
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                                 ADL32500
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                       ADL32510
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)            ADL32520
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                               ADL32530
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)              ADL32540
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)               ADL32550
     1,KBHG(3,5),EBHG(3,5)                                                ADL32560
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                    ADL32570
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)        ADL32580
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)           ADL32590
     1,IFLLDG(10)                                                         ADL32600
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                  ADL32610
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)     ADL32620
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                    ADL32630
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)      ADL32640
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)       ADL32650
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)                ADL32660
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG           ADL32670
     1,DFACG(10)                                                          ADL32680
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                          ADL32690
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                         ADL32700
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)            ADL32710
     1,INTFLG(10)                                                         ADL32720
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT          ADL32730
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                        ADL32740
C-----------------------------------------------------------------------  ADL32750
C FILE 'CHEML.COM'                                                        ADL32760
        COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)            ADL32770
        COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),        ADL32780
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                      ADL32790
        COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                         ADL32800
        COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                             ADL32810
        COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),             ADL32820
     2  CHEMPC,BIOPCT,VOLPCT,EXPPCT                                       ADL32830
        COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)                 ADL32840
        REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                         ADL32850
C-----------------------------------------------------------------------  ADL32860
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                     ADL32870
C-----------------------------------------------------------------------  ADL32880
C FILE 'ENVIRL.COM'                                                       ADL32890
        COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                    ADL32900
        COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),         ADL32910
     2  NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                             ADL32920
        COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,            ADL32930
     2  KDTIME,TFACTR,IPER,PERIOD                                         ADL32940
        COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)         ADL32950
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                         ADL32960
        COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),         ADL32970
     1    PSEDCO(10)                                                      ADL32980
        COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                  ADL32990
     2 SEDFL(10,10),WATINL(10)                                            ADL33000
        COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),              ADL33010
     1  WAVEL(39),TOTETA(39),WLAML(39)                                    ADL33020
        REAL INTFL,NPSFL,NPSCOL                                           ADL33030
        REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                    ADL33040
        LOGICAL PRINT                                                     ADL33050
        INTEGER L,E,H,B                                                   ADL33060
```

```
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C THIS VERSION DIMENSIONED FOR 10 COMPARTMENTS AT MOST:
      DIMENSION Y(10)
      DIMENSION YSAV(10),JJJ(10)
C YSAV FOR Y ON PREVIOUS ITERATION, TEST FOR CONVERGENCE TEST
      REAL INTLDL
C COUNTER FOR COMPUTATION LOOP:
      INTEGER*4 LKOUNT
C-----------------------------------------------------------------------
C DATA FOR OBJECT TIME FORMAT TO SUPPRESS PRINTING OF ZERO LOADINGS.
      DOUBLE PRECISION FMT(13),BLK1,PT1,PT2
      REAL*4 OUT(5),BLANK
      DATA FMT/8H(3X,I2,3,8HX       ,8H,2X     ,8H       ,
     18H,2X     ,8H       ,8H,2X     ,8H       ,8H,2X     ;
     28H       ,8H,2X     ,8H       ,8H)      /
      DATA BLK1,PT2 /8H,1A1,10X,8H,G11.4  /
      DATA PT1 /8H,1PG11.4/
      DATA BLANK/4H    /
C FOR FIRST LOAD COMPUTE STEADY, OTHERWISE JUST PRINT LOAD TABLE
      IF(IPER.NE.1) GO TO 230
C COUNTER TO PREVENT INFINITE LOOP IN EVENT OF PROBLEMS:
      DO 100 J=1,KOUNT
      JJJ(J)=0
100   CONTINUE
110   CONTINUE
C INITIALIZE VALUES:
      DO 115 J=1,KOUNT
      YSAV(J)=0.
      Y(J)=0.
115   CONTINUE
C-----------------------------------------------------------------------
C COMPUTE STEADY-STATE CONCENTRATIONS:
C  ALLOW FOR 100000 TRIALS:
C
      DO 150 LKOUNT=1,100000
C CONVERGENCE CRITERION (ALL CONVERGED) TEST INDEX:
     ICRIT=1
       DO 130 J=1,KOUNT
C COMPUTE INTERNAL LOADINGS:
       INTLDL=0.
      DO 120 I=1,KOUNT
        INTLDL=INTLDL+INTINL(I,J)*Y(I)
120     CONTINUE
C COMPUTE CONCENTRATION:
       Y(J)=(CONLDL(J)+INTLDL)/TOTKL(J)
C RELATIVE ERROR TESTS TO ENSURE THAT ALL COMPARTMENTS CONVERGED:
       IF (Y(J).NE.0.) TEST=1.-YSAV(J)/Y(J)
       IF(TEST.GT.RELERR) ICRIT=0
       YSAV(J)=Y(J)
130   CONTINUE
C IF ALL COMPARTMENTS CONVERGED, EXIT LOOP:
      IF(ICRIT.EQ.1) GO TO 160
150   CONTINUE
C-----------------------------------------------------------------------
C COMPLETION OF THE LOOP WITHOUT CONVERGENCE INDICATES A SEVERE PROBLEM,
C E.G. A PERSISTENT CHEMICAL IN AN ECOSYSTEM LACKING CARRIER EXPORTS
```

```
C (FOR EXAMPLE A POND WITH EVAPORATIVE WATER LOSSES ONLY).  THEREFORE   ADL33640
C AETOX1 ABORTS THE ANALYSIS AND RETURNS CONTROL TO THE USER FOR        ADL33650
C EVALUATION.                                                           ADL33660
      WRITE(TTYOUT,155)                                                 ADL33670
      IFLAG=8                                                           ADL33680
      GO TO 260                                                         ADL33690
155   FORMAT(/1X,'STEADY-STATE CONCENTRATIONS CANNOT BE COMPUTED FOR', ADL33700
     1' THIS ECOSYSTEM.'/1X,'TRY A SYSTEM THAT INCLUDES AN EXPORT'      ADL33710
     2,' FLOW.')                                                        ADL33720
C-----------------------------------------------------------------------ADL33730
160   CONTINUE                                                          ADL33740
C CHECK LOADS AND ADJUST DRIFT LOADS IF SATURATION                      ADL33750
C EXCEEDED.  START FROM COMPARTMENT 1 AND PROCEED DOWNSTREAM,           ADL33760
C RE-INVOKING THE STEADY-STATE COMPUTATION LOOP AS REQUIRED.            ADL33770
      DO 230 J=1,KOUNT                                                  ADL33780
C THERE ARE 5 CHEMICAL SPECIES TO BE CONSIDERED:                        ADL33790
C LOAD COUNTER FOR LOCATING DISTRIBUTION FRACTIONS (ALPHA):             ADL33800
      II=-2                                                             ADL33810
      DO 220 I=1,5                                                      ADL33820
C INCREMENT COUNTER FOR ALPHA ADDRESS:                                  ADL33830
      II=II+1                                                           ADL33840
C INCREMENT THE SPECIES LOOP IF THE CURRENT SPECIES DOESN'T OCCUR:      ADL33850
      IF(SPFLG(I).EQ.0) GO TO 220                                       ADL33860
C                                                                       ADL33870
C CHECK FOR SUPERSATURATION (ACTUALLY 80% OF SATURATION TO KEEP IN      ADL33880
C RANGE OF LINEAR ISOTHERMS):                                           ADL33890
      SATST=ALPHA(2*I+II,J)*Y(J)                                        ADL33900
      CHECK = 0.80 * YSATL(I,J)                                         ADL33910
      IF(SATST.LT.CHECK) GO TO 220                                      ADL33920
C SOLUBILITY LIMITATIONS EXCEEDED - REDUCE ALL DRIFT LOADS              ADL33930
C AND RE-ACTIVATE STEADY-STATE COMPUTATIONS:                            ADL33940
C 1. FACTOR FOR REDUCING LOADS:                                         ADL33950
      FCTR=SATST/CHECK                                                  ADL33960
C INCREASE FACTOR BY 15% FOR FASTER CONVERGENCE:                        ADL33970
      FCTR=FCTR*1.15                                                    ADL33980
C 2. REPORT PROBLEM TO USER:                                            ADL33990
      WRITE(TTYOUT,170) J                                               ADL34000
170   FORMAT(/1X,'SOLUBILITY CRITERIA EXCEEDED IN SYSTEM ELEMENT ',    ADL34010
     1 I3,'.')                                                          ADL34020
C 3. REDUCE DRIFT LOADS IF THEY ARE NON-ZERO:                           ADL34030
      TOTDRF=0.                                                         ADL34040
      DO 180 JJ=1,KOUNT                                                 ADL34050
       TOTDRF=TOTDRF+DRFLDG(JJ)                                         ADL34060
180   CONTINUE                                                          ADL34070
C IF NO DRIFT LOADS ARE PRESENT, USER WILL HAVE TO REDUCE OTHER         ADL34080
C LOADS BY HAND: SET FLAG FOR DELETING REPORT AND RETURN TO EXAMS       ADL34090
      IF (TOTDRF.EQ.0.) GO TO 240                                       ADL34100
C REDUCE DRIFT LOADS AND RECALCULATE TOTAL LOADINGS:                    ADL34110
      SYSLDL=0.                                                         ADL34120
      DO 190 JJ=1,KOUNT                                                 ADL34130
       DRFLDG(JJ)=DRFLDG(JJ)/FCTR                                       ADL34140
C RE-EVALUATE TOTAL LOAD:                                               ADL34150
       TOTLDL(JJ)=IFLLDG(JJ)+STRLDG(JJ)+NPSLDG(JJ)+PCPLDG(JJ)+          ADL34160
     2             DRFLDG(JJ)                                           ADL34170
       CONLDL(JJ)=1.E6*TOTLDL(JJ)/WATVOL(JJ)                            ADL34180
       SYSLDL=SYSLDL+TOTLDL(JJ)                                         ADL34190
190   CONTINUE                                                          ADL34200
```

```
C EXIT THE LOOP AND RE-DO THE STEADY-STATE COMPUTATIONS:                ADL34210
      WRITE(TTYOUT,200) FCTR                                            ADL34220
200   FORMAT(/1X,'DRIFT LOADS HAVE BEEN REDUCED BY A FACTOR OF ',       ADL34230
     1 1PG11.4/)                                                        ADL34240
C EXIT THE LOOP:                                                        ADL34250
      JJJ(J)=JJJ(J)+1                                                   ADL34260
      IF(JJJ(J).LE.10) GO TO 110                                        ADL34270
C IF ALL THE OTHER LOADS HAVE BEEN SET UP AT SATURATION, THIS COULD     ADL34280
C BE A FRUITLESS EXERCISE ( AN NP*NP*10 LOOP) THEREFORE:                ADL34290
C (NP IS NUMBER OF COMPARTMENTS AVAILABLE, IN THIS VERSION NP=10)       ADL34300
      WRITE(TTYOUT,210)                                                 ADL34310
210   FORMAT(/1X,'HOWEVER, THIS DOES NOT SEEM TO BE WORKING:')          ADL34320
      GO TO 240                                                         ADL34330
C                                                                       ADL34340
220   CONTINUE                                                          ADL34350
230   CONTINUE                                                          ADL34360
C-----------------------------------------------------------------------ADL34370
C ACCEPTABLE STEADY-STATE VALUES COMPUTED:                              ADL34380
C TABULATE LOADS ON SYSTEM, PRINT IN REPORT.DAT -                       ADL34390
C TABLE TITLE AND COLUMN HEADERS:                                       ADL34400
      IF (PRINT) WRITE(RPTOUT,232)                                      ADL34410
232   FORMAT(15X,'TOXICANT LOADINGS (KG/HR) BY SYSTEM ELEMENT.')        ADL34420
      IF (PRINT) WRITE(RPTOUT,233)                                      ADL34430
233   FORMAT(1X,72(1H-)/1X,'ELEMENT',2X,'STREAM FLOW',4X,'RAINFALL',    ADL34440
     2 4X,'INTERFLOW',5X,'NPS LOAD',4X,'DRIFT LOAD'/1X,7(1H-),2X,       ADL34450
     3 11(1H-),4(3X,10(1H-)))                                           ADL34460
      DO 238 J=1,KOUNT                                                  ADL34470
      OUT(1)=STRLDG(J)                                                  ADL34480
      OUT(2)=PCPLDG(J)                                                  ADL34490
      OUT(3)=IFLLDG(J)                                                  ADL34500
      OUT(4)=NPSLDG(J)                                                  ADL34510
      OUT(5)=DRFLDG(J)                                                  ADL34520
C BUILD FORMAT STATEMENT:                                               ADL34530
      KKL=0                                                             ADL34540
      DO 237 I=1,5                                                      ADL34550
      IF (OUT(I).NE.0.0) GO TO 235                                      ADL34560
C LOAD IS ZERO:                                                         ADL34570
      FMT(2*I+2)=BLK1                                                   ADL34580
      OUT(I)=BLANK                                                      ADL34590
      GO TO 237                                                         ADL34600
235   CONTINUE                                                          ADL34610
C LOAD IS NON-ZERO                                                      ADL34620
      KKL=KKL+1                                                         ADL34630
      FMT(2*I+2)=PT2                                                    ADL34640
      IF (KKL.EQ.1) FMT(2*I+2) = PT1                                    ADL34650
237   CONTINUE                                                          ADL34660
      IF (PRINT) WRITE(RPTOUT,FMT) J, OUT                               ADL34670
238   CONTINUE                                                          ADL34680
      IF (PRINT) WRITE(RPTOUT,239)                                      ADL34690
239   FORMAT(1X,72(1H-))                                                ADL34700
C                                                                       ADL34710
C-----------------------------------------------------------------------ADL34720
      GO TO 260                                                         ADL34730
240   IFLAG=8                                                           ADL34740
      WRITE(TTYOUT,250)                                                 ADL34750
250   FORMAT(/3X,'SOLUBILITY CRITERIA EXCEEDED, CAUSE NOT APPARENT.'/   ADL34760
     1 3X,'PLEASE REDUCE LOADS AND TRY AGAIN.')                         ADL34770
```

```
260     CONTINUE
        RETURN
        END
C FILE 'AVEOUT.FTN'
        SUBROUTINE AVEOUT(Y)
C-----------------------------------------------------------------------
C AVEOUT. AVEOUT COMPUTES AND REPORTS THE AVERAGE, MAXIMUM, AND MINIMUM
C STEADY-STATE CONCENTRATIONS OF TOXICANT.  AVEOUT ALSO GIVES AN ELEMENT
C BY ELEMENT LISTING OF THE CONCENTRATIONS AND DISTRIBUTION OF TOXICANT
C THROUGHOUT THE ECOSYSTEM.  AVEOUT WRITES TO 'REPORT.DAT' AND TO A
C FILE USED FOR PLOTTING THE RESULTS ('EXPO.DAT').
C------------------------------------------------------------------------
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG
     1,NPSEDG,INTFLG
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT
      LOGICAL*1 CHEMNA,SYSTYP
      COMMON/NAME1G/CHEMNA(60)
      COMMON/CONTRG/SPFLG(5)
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)
     1,KBHG(3,5),EBHG(3,5)
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)
     1,IFLLDG(10)
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG
     1,DFACG(10)
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)
     1,INTFLG(10)
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:
C-----------------------------------------------------------------------
C FILE 'CHEML.COM'
        COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)
        COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)
        COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)
      COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                             ADL35350
      COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),             ADL35360
     2  CHEMPC,BIOPCT,VOLPCT,EXPPCT                                     ADL35370
      COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)                 ADL35380
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                         ADL35390
C-----------------------------------------------------------------------ADL35400
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRCNMENT:                   ADL35410
C-----------------------------------------------------------------------ADL35420
C FILE 'ENVIRL.COM'                                                     ADL35430
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                    ADL35440
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),         ADL35450
     2  NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                           ADL35460
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,            ADL35470
     2  KDTIME,TFACTR,IPER,PERIOD                                       ADL35480
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)         ADL35490
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL35500
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),         ADL35510
     1     PSEDCO(10)                                                   ADL35520
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                  ADL35530
     2 SEDFL(10,10),WATINL(10)                                          ADL35540
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),              ADL35550
     1  WAVEL(39),TOTETA(39),WLAML(39)                                  ADL35560
C**********                                                             ADL35570
      LOGICAL NPS,DOT                                                   ADL35580
      COMMON/NPSBLK/NPS,LUNPS,LUNPS2,NPSPER                             ADL35590
      COMMON/DOTBLK/DOT,LUDOT                                           ADL35600
C**********                                                             ADL35610
      REAL INTFL,NPSFL,NPSCOL                                           ADL35620
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                    ADL35630
      LOGICAL PRINT                                                     ADL35640
      INTEGER L,E,H,B                                                   ADL35650
C-----------------------------------------------------------------------ADL35660
C-----------------------------------------------------------------------ADL35670
C SET UP CODE FOR TRANSFORMATION OF COMPARTMENT TYPES TO LOGICAL*1      ADL35680
C FOR WRITING TO UNFORMATTED PLOT FILES:                                ADL35690
      DIMENSION COMTYP(10)                                              ADL35700
C                   ALLOWING FOR A  MAXIMUM OF 10 COMPARTMENTS.         ADL35710
      LOGICAL*1 COMTYP,LIT,EPIL,HYPOL,BENTH                             ADL35720
C-----------------------------------------------------------------------ADL35730
      DIMENSION Y(1)                                                    ADL35740
C Y IS CONSTITUENT CONCENTRATION (MG/L) REFERRED TO                     ADL35750
C AQUEOUS PHASE OF COMPARTMENT.                                         ADL35760
      LOGICAL*1 STAR                                                    ADL35770
C WRITE STEADY-STATE CONCENTRATION (WITH NAME OF CHEMICAL AND ECOSYSTEM)ADL35780
C TO FILE FOR PLOTTING CONCENTRATION VS. ECOSYSTEM,                     ADL35790
C AND REPORT STEADY-STATE DISTRIBUTION TO REPORT.DAT.                   ADL35800
C                                                                       ADL35810
C "STAR" IS DELIMITER USED IN UNFORMATTED PLOTTING FILES                ADL35820
      DATA STAR/1H*/                                                    ADL35830
C-----------------------------------------------------------------------ADL35840
C DATA FOR TRANSFORMING COMPARTMENT TYPE TO L*1 FOR PLOT FILES:         ADL35850
      DATA LIT/1HL/,EPIL/1HE/,HYPOL/1HH/,BENTH/1HB/                     ADL35860
C-----------------------------------------------------------------------ADL35870
C                                                                       ADL35880
C                                                                       ADL35890
C**********                                                             ADL35900
      IF (DOT.OR.NPS) RETURN                                            ADL35910
```

```
C**********                                                             ADL35920
C                                                                       ADL35930
C                                                                       ADL35940
C LOAD LOGICAL*1 COMPARTMENT TYPES FOR WRITING TO UNFORMATTED PLOT FILE:ADL35950
      DO 5 J=1,KOUNT                                                    ADL35960
      IF(TYPEE(J).EQ.L) COMTYP(J)=LIT                                   ADL35970
      IF(TYPEE(J).EQ.E) COMTYP(J)=EPIL                                  ADL35980
      IF(TYPEE(J).EQ.H) COMTYP(J)=HYPOL                                 ADL35990
      IF(TYPEE(J).EQ.B) COMTYP(J)=BENTH                                 ADL36000
5     CONTINUE                                                          ADL36010
C-----------------------------------------------------------------------ADL36020
C                                                                       ADL36030
C  ZERO 'Z' VALUES:                                                     ADL36040
      DO 10 I=1,18                                                      ADL36050
              Z(I)=0.                                                   ADL36060
10    CONTINUE                                                          ADL36070
C                                                                       ADL36080
C ALLOW FOR RUN OF A SINGLE WATER COLUMN COMPARTMENT:                   ADL36090
      IF(KOUNTS.EQ.0) KOUNTS=1                                          ADL36100
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ADL36110
C                                                                       ADL36120
C  COMPUTE MEAN CONCENTRATIONS:                                         ADL36130
C                                                                       ADL36140
C DO COMPARTMENTS IN NUMERICAL ORDER:                                   ADL36150
      DO 12 J=1,KOUNT                                                   ADL36160
C                                  BOTTOM SEDIMENT:                     ADL36170
         IF(INDEXS(J).EQ.1) GO TO 11                                    ADL36180
C WATER COLUMN COMPARTMENTS:                                            ADL36190
C DISSOLVED:                                                            ADL36200
      Z(1)=Z(1)+ALPHA(16,J)*Y(J)                                        ADL36210
C SORBED MG/KG:                                                         ADL36220
      Z(2)=Z(2)+ALPHA(17,J)*Y(J)/SEDCOL(J)                              ADL36230
C MASS, KG (1.E-6 KG/MG)                                                ADL36240
      Z(5)=Z(5)+Y(J)*WATVOL(J)*1.E-6                                    ADL36250
      GO TO 12                                                          ADL36260
C                                                                       ADL36270
C BOTTOM SEDIMENTS:                                                     ADL36280
11    CONTINUE                                                          ADL36290
C MG/L DISSOLVED IN PORE WATER:                                         ADL36300
      Z(3)=Z(3)+ALPHA(16,J)*Y(J)                                        ADL36310
C SORBED MG/KG                                                          ADL36320
      Z(4)=Z(4)+ALPHA(17,J)*Y(J)/SEDCOL(J)                              ADL36330
C POLLUTANT MASS IN KG (1.E-6 KG/MG)                                    ADL36340
      Z(6)=Z(6)+Y(J)*WATVOL(J)*1.E-6                                    ADL36350
12    CONTINUE                                                          ADL36360
C                                                                       ADL36370
C COMPUTE AVERAGES IN WATER COLUMN:                                     ADL36380
C MG/L  DISSOLVED                                                       ADL36390
      Z(1)=Z(1)/KOUNTW                                                  ADL36400
C MG/KG SORBED:                                                         ADL36410
      Z(2)=Z(2)/KOUNTW                                                  ADL36420
C                                                                       ADL36430
C COMPUTE AVERAGES IN BOTTOM SEDIMENTS:                                 ADL36440
C MG/L DISSOLVED:                                                       ADL36450
      Z(3)=Z(3)/KOUNTS                                                  ADL36460
C MG/KG SORBED                                                          ADL36470
      Z(4)=Z(4)/KOUNTS                                                  ADL36480
```

```
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ADL36490
C                                                                          ADL36500
C INITIALIZE MAX AND MIN TRACKERS AND POINTERS:                            ADL36510
      DO 100 J=1,10                                                        ADL36520
      DOMAX(J)=0.                                                          ADL36530
      DOMIN(J)=1.E25                                                       ADL36540
      MAXPT(J)=0                                                           ADL36550
      MINPT(J)=0                                                           ADL36560
100   CONTINUE                                                             ADL36570
C WRITE PAGE HEADER TO REPORT.DAT:                                         ADL36580
      WRITE(RPTOUT,110) CHEMNA,SYSTYP                                      ADL36590
110   FORMAT(1H1,'  AERL-ESB MODEL OF FATE OF ORGANIC TOXICANTS',         ADL36600
     11X,'IN AQUATIC ECOSYSTEMS.'/1X,'CHEMICAL: ',60A1/1X,                ADL36610
     2'ECOSYSTEM: ',60A1/1X,72(1H-))                                      ADL36620
      WRITE(RPTOUT,120)                                                    ADL36630
120   FORMAT(1X,'DISTRIBUTION OF ORGANIC TOXICANT AT STEADY',             ADL36640
     1 '-STATE:  IN THE WATER COLUMN:')                                    ADL36650
C                                                                          ADL36660
      WRITE(RPTOUT,130)                                                    ADL36670
C DASHED LINE FORMAT STATEMENT:                                            ADL36680
130   FORMAT(1X,72(1H-))                                                   ADL36690
C SET UP TABLE IN REPORT.DAT:                                              ADL36700
C                                                                          ADL36710
C     COLUMN HEADERS FOR DISTRIBUTION TABLE:                               ADL36720
      WRITE(RPTOUT,140)                                                    ADL36730
140   FORMAT(1X,'COMP STEADY-STATE RESIDENT MASS',4X,4(1H*),1X,           ADL36740
     1 'TOXICANT CONCENTRATIONS',1X,4(1H*))                                ADL36750
      WRITE(RPTOUT,150)                                                    ADL36760
150   FORMAT(2X,'#',6X,'2',25X,'TOTAL',3X,'DISSOLVED',                     ADL36770
     1 1X,'SEDIMENTS',3X,'BIOTA')                                          ADL36780
C                                                                          ADL36790
      WRITE(RPTOUT,160)                                                    ADL36800
160   FORMAT(6X,'G/M',7X,'KILOS',8X,'%',6X,'MG/*',5X,                      ADL36810
     1 'MG/L',6X,'MG/KG',6X,'UG/G')                                        ADL36820
C                                                                          ADL36830
      WRITE(RPTOUT,170)                                                    ADL36840
170   FORMAT(1X,3(1H-),2X,5(1H-),4X,6(1H-),5X,6(1H-),4(1X,9(1H-)))        ADL36850
180   FORMAT(/34X,'WATER COLUMN')                                          ADL36860
C                                                                          ADL36870
C                                                                          ADL36880
C COMPUTATIONS FOR SYS. ELEMENTS: COMPUTE AND PRINT VARIABLES,             ADL36890
C ACCUMULATE DATA FOR MEAN VALUES, TRACK MAXIMA AND MINIMA                  ADL36900
C                                                                          ADL36910
C WATER COLUMN:                                                            ADL36920
C                                                                          ADL36930
      DO 290 J=1,KOUNT                                                     ADL36940
C SKIP SEDIMENT COMPARTMENTS:                                              ADL36950
        IF(INDEXW(J).EQ.0) GO TO 290                                       ADL36960
C COMPUTE MASS BY SYSTEM ELEMENT (IN KILOGRAMS):                           ADL36970
      Z(12)=1.E-6*Y(J)*WATVOL(J)                                           ADL36980
C CONVERT MASS TO GRAMS/SQUARE METER                                       ADL36990
      Z(11)=1.E3*Z(12)/AREAG(J)                                            ADL37000
C TRACK MAXIMUM VALUE                                                      ADL37010
      IF(DOMAX(1).GT.Z(11)) GO TO 190                                      ADL37020
C NEW MAXIMUM:                                                             ADL37030
      DOMAX(1)=Z(11)                                                       ADL37040
                                                                           ADL37050
```

```
C LOCATION OF MAXIMUM                                                   ADL37060
        MAXPT(1)=J                                                      ADL37070
C TRACK MINIMUM                                                         ADL37080
190     IF(DOMIN(1).LT.Z(11)) GO TO 200                                 ADL37090
C NEW MINIMUM:                                                          ADL37100
        DOMIN(1)=Z(11)                                                  ADL37110
C LOCATION OF MINIMUM:                                                  ADL37120
        MINPT(1)=J                                                      ADL37130
C ACCUMULATE FOR MEAN VALUE:                                            ADL37140
200     Z(13)=Z(13)+Z(11)                                               ADL37150
C CALCULATE MASS AS PERCENT OF TOTAL IN WATER COLUMN:                   ADL37160
        PERCET=Z(12)*100./Z(5)                                          ADL37170
C                                                                       ADL37180
C TOTAL CONCENTRATION (MG/L IN WATER COLUMN)                            ADL37190
        Z(7)=Y(J)                                                       ADL37200
C TRACK MAXIMUM VALUE                                                   ADL37210
        IF(DOMAX(2).GT.Z(7)) GO TO 210                                  ADL37220
C NEW MAXIMUM:                                                          ADL37230
        DOMAX(2)=Z(7)                                                   ADL37240
C LOCATION OF NEW MAXIMUM:                                              ADL37250
        MAXPT(2)=J                                                      ADL37260
C TRACK MINIMUM                                                         ADL37270
210     IF(DOMIN(2).LT.Z(7)) GO TO 220                                  ADL37280
C NEW MINIMUM:                                                          ADL37290
        DOMIN(2)=Z(7)                                                   ADL37300
C LOCATION:                                                             ADL37310
        MINPT(2)=J                                                      ADL37320
C ACCUMULATE FOR MEAN VALUE                                             ADL37330
220     Z(14)=Z(14)+Z(7)                                                ADL37340
C NO PERCENTAGE - -                                                     ADL37350
C                                                                       ADL37360
C FREE (DISSOLVED) TOXICANT IN WATER COLUMN:                            ADL37370
        Z(8)=ALPHA(16,J)*Y(J)                                           ADL37380
C TRACK MAXIMUM VALUE                                                   ADL37390
        IF(DOMAX(3).GT.Z(8)) GO TO 230                                  ADL37400
        DOMAX(3)=Z(8)                                                   ADL37410
        MAXPT(3)=J                                                      ADL37420
C TRACK MINIMUM                                                         ADL37430
230     IF(DOMIN(3).LT.Z(8)) GO TO 240                                  ADL37440
        DOMIN(3)=Z(8)                                                   ADL37450
C LOCATION OF NEW MINIMUM:                                              ADL37460
        MINPT(3)=J                                                      ADL37470
C MEAN ALREADY AVAILABLE AS Z(1)                                        ADL37480
C                                                                       ADL37490
C TOXICANT SORBED ON SUSPENDED SEDIMENTS (MG/KG)                        ADL37500
240     Z(9)=ALPHA(17,J)*Y(J)/SEDCOL(J)                                 ADL37510
C TRACK MAXIMUM VALUE                                                   ADL37520
        IF(DOMAX(4).GT.Z(9)) GO TO 250                                  ADL37530
        DOMAX(4)=Z(9)                                                   ADL37540
        MAXPT(4)=J                                                      ADL37550
C TRACK MINIMUM                                                         ADL37560
250     IF(DOMIN(4).LT.Z(9)) GO TO 260                                  ADL37570
        DOMIN(4)=Z(9)                                                   ADL37580
        MINPT(4)=J                                                      ADL37590
C MEAN ALREADY AVAILABLE AS Z(2)                                        ADL37600
C                                                                       ADL37610
C TOXICANT SORBED ON BIOTA (BACTERIA +), UG/G                           ADL37620
```

```
260     Z(10)=ALPHA(18,J)*Y(J)/BIOTOL(J)                                   ADL37630
C TRACK MAXIMUM VALUE                                                      ADL37640
        IF(DOMAX(5).GT.Z(10)) GO TO 270                                    ADL37650
        DOMAX(5)=Z(10)                                                     ADL37660
        MAXPT(5)=J                                                         ADL37670
C TRACK MINIMUM                                                            ADL37680
270     IF(DOMIN(5).LT.Z(10)) GO TO 280                                    ADL37690
        DOMIN(5)=Z(10)                                                     ADL37700
        MINPT(5)=J                                                         ADL37710
C ACCUMULATE SUM FOR MEAN VALUE                                            ADL37720
280     Z(15)=Z(15)+Z(10)                                                  ADL37730
C                                                                          ADL37740
C WRITE LINE TO REPORT.DAT:                                                ADL37750
        WRITE(RPTOUT,310) J,Z(11),Z(12),PERCET,(Z(I),I=7,10)               ADL37760
C WRITE LINE TO EXPO.DAT:                                                  ADL37770
        WRITE(SSOUT) J,COMTYP(J),Z(11),Z(12),PERCET,(Z(I),I=7,10)          ADL37780
290     CONTINUE                                                           ADL37790
C END OF WATER COLUMN LOOP                                                 ADL37800
C                                                                          ADL37810
C MARK PLOT FILE FOR END OF WATER COLUMN VALUES:                           ADL37820
        WRITE(SSOUT) J,STAR,(Z(I),I=1,7)                                   ADL37830
C                                                                          ADL37840
310     FORMAT(1X,I2,1X,1PG10.3,G11.4,1X,0PF6.2,1X,1PG10.3,3G10.3)         ADL37850
C                                                                          ADL37860
C COMPUTE AVERAGE VALUES WHERE NEEDED:                                     ADL37870
C  1. MASS (GRAMS/SQUARE METER)                                            ADL37880
        Z(13)=Z(13)/KOUNTW                                                 ADL37890
C  2. TOTAL CONCENTRATION                                                  ADL37900
        Z(14)=Z(14)/KOUNTW                                                 ADL37910
C  3. CONCENTRATION OF TOXICANT SORBED ON BIOTA:                           ADL37920
        Z(15)=Z(15)/KOUNTW                                                 ADL37930
C                                                                          ADL37940
C WATER COLUMN MASS AS PERCENTAGE OF TOTAL MASS IN SYSTEM:                 ADL37950
        PERCET=Z(5)*100./(Z(5)+Z(6))                                       ADL37960
C                                                                          ADL37970
C WRITE SUBTOTAL TO REPORT.DAT:                                            ADL37980
        WRITE(RPTOUT,320) Z(5),PERCET                                      ADL37990
320     FORMAT(2X,'SUBTOTAL:',3X,1PG11.4,1X,0PF6.2)                        ADL38000
C                                                                          ADL38010
C SEDIMENT COMPARTMENTS:                                                   ADL38020
C                                                                          ADL38030
C LABEL SEDIMENT SECTION IN REPORT.DAT:                                    ADL38040
        WRITE(RPTOUT,330)                                                  ADL38050
330     FORMAT(1X,' AND IN THE BOTTOM SEDIMENTS:')                         ADL38060
C                                                                          ADL38070
C COMPUTATIONS FOR BOTTOM SEDIMENTS: COMPUTE AND PRINT VARIABLES,          ADL38080
C ACCUMULATE VALUES FOR MEANS, TRACK MAXIMA AND MINIMA                     ADL38090
C                                                                          ADL38100
        DO 440 J=1,KOUNT                                                   ADL38110
C SKIP WATER COLUMN COMPARTMENTS:                                          ADL38120
        IF(INDEXS(J).EQ.0) GO TO 440                                       ADL38130
C COMPUTE MASS (KILOGRAMS) BY SYSTEM ELEMENT:                              ADL38140
        Z(12)=1.E-6*Y(J)*WATVOL(J)                                         ADL38150
C CONVERT MASS TO GRAMS/SQUARE METER                                       ADL38160
        Z(11)=1.E3*Z(12)/AREAG(J)                                          ADL38170
C TRACK MAXIMUM                                                            ADL38180
        IF(DOMAX(6).GT.Z(11)) GO TO 340                                    ADL38190
        DOMAX(6)=Z(11)                                                  ADL38200
        MAXPT(6)=J                                                      ADL38210
C TRACK MINIMUM                                                         ADL38220
340     IF (DOMIN(6).LT.Z(11)) GO TO 350                                ADL38230
        DOMIN(6)=Z(11)                                                  ADL38240
        MINPT(6)=J                                                      ADL38250
C ACCUMULATE FOR AVERAGE                                                ADL38260
350     Z(16)=Z(16)+Z(11)                                               ADL38270
C CALCULATE MASS AS A PERCENTAGE OF TOTAL IN BOTTOM SEDIMENTS           ADL38280
        PERCET=Z(12)*100./Z(6)                                          ADL38290
C TOTAL CONCENTRATION, MG/KG                                            ADL38300
C    (DIVISION BY SEDMSL IS ALL RIGHT BECAUSE SEDIMENTS IN A            ADL38310
C     'BOTTOM SEDIMENT' COMPARTMENT ARE NEVER ZERO)                     ADL38320
        Z(7)=Y(J)*WATVOL(J)/SEDMSL(J)                                   ADL38330
C TRACK MAXIMUM VALUE                                                   ADL38340
        IF(DOMAX(7).GT.Z(7)) GO TO 360                                  ADL38350
        DOMAX(7)=Z(7)                                                   ADL38360
        MAXPT(7)=J                                                      ADL38370
C TRACK MINIMUM                                                         ADL38380
360     IF(DOMIN(7).LT.Z(7)) GO TO 370                                  ADL38390
        DOMIN(7)=Z(7)                                                   ADL38400
        MINPT(7)=J                                                      ADL38410
C ACCUMULATE FOR AVERAGE VALUE                                          ADL38420
370     Z(17)=Z(17)+Z(7)                                                ADL38430
C NO PERCENTAGES                                                        ADL38440
C                                                                       ADL38450
C FREE (DISSOLVED) TOXICANT IN BOTTOM SEDIMENTS                         ADL38460
C (MG/L OF INTERSTITIAL WATER)                                          ADL38470
        Z(8)=ALPHA(16,J)*Y(J)                                           ADL38480
C TRACK MAXIMUM VALUE                                                   ADL38490
        IF(DOMAX(8).GT.Z(8)) GO TO 380                                  ADL38500
        DOMAX(8)=Z(8)                                                   ADL38510
        MAXPT(8)=J                                                      ADL38520
C TRACK MINIMUM                                                         ADL38530
380     IF(DOMIN(8).LT.Z(8)) GO TO 390                                  ADL38540
        DOMIN(8)=Z(8)                                                   ADL38550
        MINPT(8)=J                                                      ADL38560
C MEAN ALREADY AVAILABLE AS Z(3)                                        ADL38570
C                                                                       ADL38580
C SORBED ON BOTTOM SEDIMENTS:                                           ADL38590
390     Z(9)=ALPHA(17,J)*Y(J)/SEDCOL(J)                                 ADL38600
C TRACK MAXIMUM VALUE                                                   ADL38610
        IF(DOMAX(9).GT.Z(9)) GO TO 400                                  ADL38620
        DOMAX(9)=Z(9)                                                   ADL38630
        MAXPT(9)=J                                                      ADL38640
C TRACK MINIMUM                                                         ADL38650
400     IF(DOMIN(9).LT.Z(9)) GO TO 410                                  ADL38660
        DOMIN(9)=Z(9)                                                   ADL38670
        MINPT(9)=J                                                      ADL38680
C MEAN VALUE ALREADY AVAILABLE AS Z(4)                                  ADL38690
C                                                                       ADL38700
C TOXICANT SORBED ON BENTHIC ORGANISMS (UG/G):                          ADL38710
410     Z(10)=ALPHA(18,J)*Y(J)/BIOTOL(J)                                ADL38720
C TRACK MAXIMUM VALUE                                                   ADL38730
        IF(DOMAX(10).GT.Z(10)) GO TO 420                                ADL38740
        DOMAX(10)=Z(10)                                                 ADL38750
        MAXPT(10)=J                                                     ADL38760
```

```
C TRACK MINIMUM VALUE                                                   ADL38770
420     IF(DOMIN(10).LT.Z(10)) GO TO 430                                ADL38780
        DOMIN(10)=Z(10)                                                 ADL38790
        MINPT(10)=J                                                     ADL38800
C ACCUMULATE SUM FOR MEAN VALUE                                         ADL38810
430     Z(18)=Z(18)+Z(10)                                               ADL38820
C                                                                       ADL38830
C WRITE LINE TO REPORT.DAT:                                             ADL38840
        WRITE(RPTOUT,310) J,Z(11),Z(12),PERCET,(Z(I),I=7,10)            ADL38850
C WRITE LINE TO EXPO.DAT:                                               ADL38860
        WRITE(SSOUT) J,COMTYP(J),Z(11),Z(12),PERCET,(Z(I),I=7,10)       ADL38870
C                                                                       ADL38880
440     CONTINUE                                                        ADL38890
C END OF BOTTOM SEDIMENT LOOP.                                          ADL38900
C                                                                       ADL38910
C MARK PLOT FILE FOR END OF BOTTOM SEDIMENTS:                           ADL38920
        WRITE(SSOUT) J,STAR,(Z(I),I=1,7)                                ADL38930
C                                                                       ADL38940
C                                                                       ADL38950
C COMPUTE AVERAGES WHERE NEEDED:                                        ADL38960
C   1. MASS (GRAMS/SQUARE METER)                                        ADL38970
        Z(16)=Z(16)/KOUNTS                                              ADL38980
C   2. TOTAL CONCENTRATION                                              ADL38990
        Z(17)=Z(17)/KOUNTS                                              ADL39000
C   3. CONCENTRATION OF TOXICANT SORBED ON BENTHOS:                     ADL39010
        Z(18)=Z(18)/KOUNTS                                              ADL39020
C                                                                       ADL39030
C BOTTOM SEDIMENT MASS AS PERCENT OF TOTAL MASS:                        ADL39040
        PERCET=Z(6)*100./(Z(5)+Z(6))                                    ADL39050
C                                                                       ADL39060
C WRITE SUBTOTAL TO REPORT.DAT:                                         ADL39070
        WRITE(RPTOUT,320) Z(6),PERCET                                   ADL39080
C                                                                       ADL39090
C WRITE TOTAL MASS TO REPORT.DAT:                                       ADL39100
        TOTMAS=Z(5)+Z(6)                                                ADL39110
        WRITE(RPTOUT,450) TOTMAS                                        ADL39120
450     FORMAT(1X,'TOTAL MASS (KILOGRAMS) ='3X,1PG11.4)                 ADL39130
C                                                                       ADL39140
        WRITE(RPTOUT,130)                                               ADL39150
C                                                                       ADL39160
        WRITE(RPTOUT,460)                                               ADL39170
460     FORMAT(1X,'* TOTAL CONCENTRATION AS MG/L IN WATER',             ADL39180
     1  ' COLUMN, AS MG/KG IN SEDIMENTS.')                              ADL39190
C WRITE OUT TABLE OF AVERAGES, MAXIMA AND MINIMA:                       ADL39200
C                                                                       ADL39210
C WRITE PAGE HEADER                                                     ADL39220
        WRITE(RPTOUT,110) CHEMNA,SYSTYP                                 ADL39230
C TITLE FOR TABLE:                                                      ADL39240
        WRITE(RPTOUT,470)                                               ADL39250
470     FORMAT(1X,'AVERAGE, MAXIMUM AND MINIMUM CONCENTRATIONS OF',     ADL39260
     1' TOXICANT AT STEADY-STATE.')                                     ADL39270
        WRITE(RPTOUT,130)                                               ADL39280
C SET UP TABLE COLUMNS:                                                 ADL39290
        WRITE(RPTOUT,480)                                               ADL39300
480     FORMAT(3X,'1'/4X,'(CP)TOTAL',5X,'(CP)DISSOLVED (CP)',           ADL39310
     1'SEDIMENTS (CP)BIOTA',5X,'(CP)MASS'/68X,'2'/10X,                  ADL39320
     2'MG/*',10X,'MG/L',9X,'MG/KG',9X,'UG/G',10X,'G/M')                 ADL39330
```

```
        WRITE(RPTOUT,490)                                               ADL39340
490     FORMAT(4X,11(1H-),4(3X,11(1H-))//1X,'WATER COLUMN:')            ADL39350
        WRITE(RPTOUT,500) Z(14),Z(1),Z(2),Z(15),Z(13)                   ADL39360
500     FORMAT(1X,'AV',1X,4(4X,1PG10.3),4X,G9.2)                        ADL39370
        WRITE(SSOUT) Z(14),Z(1),Z(2),Z(15),Z(13)                        ADL39380
        WRITE(RPTOUT,510) MAXPT(2),DOMAX(2),MAXPT(3),DOMAX(3),MAXPT(4), ADL39390
     1 DOMAX(4),MAXPT(5),DOMAX(5),MAXPT(1),DOMAX(1)                     ADL39400
510     FORMAT(1X,'MA'1X,4('(',I2,')',1PG10.3),'(',I2,')',G9.2)         ADL39410
        WRITE(SSOUT) (MAXPT(I),I=2,5),MAXPT(1),(DOMAX(I),I=2,5),DOMAX(1)ADL39420
        WRITE(RPTOUT,520) MINPT(2),DOMIN(2),MINPT(3),DOMIN(3),MINPT(4), ADL39430
     1 DOMIN(4),MINPT(5),DOMIN(5),MINPT(1),DOMIN(1)                     ADL39440
520     FORMAT(1X,'MI',1X,4('(',I2,')',1PG10.3),'(',I2,')',G9.2)        ADL39450
        WRITE(SSOUT) (MINPT(I),I=2,5),MINPT(1),(DOMIN(I),I=2,5),DOMIN(1)ADL39460
C                                                                       ADL39470
        WRITE(RPTOUT,530)                                               ADL39480
530     FORMAT(/1X,'BOTTOM SEDIMENTS:')                                 ADL39490
        WRITE(RPTOUT,500) Z(17),Z(3),Z(4),Z(18),Z(16)                   ADL39500
        WRITE(SSOUT) Z(17),Z(3),Z(4),Z(18),Z(16)                        ADL39510
C                                                                       ADL39520
        WRITE(RPTOUT,510) MAXPT(7),DOMAX(7),MAXPT(8),DOMAX(8),MAXPT(9), ADL39530
     1 DOMAX(9),MAXPT(10),DOMAX(10),MAXPT(6),DOMAX(6)                   ADL39540
      WRITE(SSOUT) (MAXPT(I),I=7,10),MAXPT(6),(DOMAX(I),I=7,10),DOMAX(6)ADL39550
        WRITE(RPTOUT,520) MINPT(7),DOMIN(7),MINPT(8),DOMIN(8),MINPT(9), ADL39560
     1 DOMIN(9),MINPT(10),DOMIN(10),MINPT(6),DOMIN(6)                   ADL39570
      WRITE(SSOUT) (MINPT(I),I=7,10),MINPT(6),(DOMIN(I),I=7,10),DOMIN(6)ADL39580
        WRITE(RPTOUT,130)                                               ADL39590
        WRITE(RPTOUT,540)                                               ADL39600
540     FORMAT(1X,'1 NUMBER IN PARENS (CP) INDICATES COMPARTMENT ',     ADL39610
     1  'WHERE VALUE WAS FOUND.')                                       ADL39620
C UNITS FOOTNOTE TO TABLE:                                              ADL39630
        WRITE(RPTOUT,460)                                               ADL39640
C                                                                       ADL39650
        RETURN                                                          ADL39660
      END                                                               ADL39670
C FILE 'FLXOUT.FTN'                                                     ADL39680
        SUBROUTINE FLXOUT(Y)                                            ADL39690
C---------------------------------------------------------------------- ADL39700
C FLXOUT. FLXOUT ANALYZES AND REPORTS THE STEADY-STATE FATE OF THE      ADL39710
C TOXICANT.  THE INFORMATION REPORTED FOR EACH PROCESS INCLUDES THE     ADL39720
C MASS FLUX (KG/TIME) ATTRIBUTABLE TO THE PROCESS, THE PERCENTAGE OF    ADL39730
C LOAD CONSUMED IN THE PROCESS, AND A PROJECTED HALF-LIFE THAT WOULD    ADL39740
C RESULT FROM EACH PROCESS ACTING IN ISOLATION.  THE PROCESSES ARE ALSO ADL39750
C SUMMED BY CATEGORY (CHEMICAL, BIOLOGICAL, TRANSPORT), AND A TOTAL     ADL39760
C SYSTEM HALF-LIFE IS ESTIMATED FROM THE TOTAL FLUX IN ORDER TO SPECIFY ADL39770
C A TIME-FRAME FOR THE INTEGRATION ROUTINE ('RKF45').                   ADL39780
C-----------------------------------------------------------------------ADL39790
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL39800
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL39810
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL39820
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL39830
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG     ADL39840
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL39850
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL39860
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL39870
     1,NPSEDG,INTFLG                                                    ADL39880
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH     ADL39890
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL39900
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL39910
      COMMON/NAME1G/CHEMNA(60)                                          ADL39920
      COMMON/CONTRG/SPFLG(5)                                            ADL39930
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL39940
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL39950
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL39960
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL39970
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL39980
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL39990
     1,KBHG(3,5),EBHG(3,5)                                              ADL40000
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL40010
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL40020
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL40030
     1,IFLLDG(10)                                                       ADL40040
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL40050
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL40060
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL40070
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL40080
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)     ADL40090
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)              ADL40100
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG         ADL40110
     1,DFACG(10)                                                        ADL40120
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                        ADL40130
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                       ADL40140
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)          ADL40150
     1,INTFLG(10)                                                       ADL40160
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL40170
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                      ADL40180
C-----------------------------------------------------------------------ADL40190
C FILE 'CHEML.COM'                                                      ADL40200
        COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)          ADL40210
        COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),      ADL40220
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                    ADL40230
        COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                       ADL40240
        COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                           ADL40250
        COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),           ADL40260
     2  CHEMPC,BIOPCT,VOLPCT,EXPPCT                                     ADL40270
        COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)               ADL40280
        REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                       ADL40290
C-----------------------------------------------------------------------ADL40300
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                   ADL40310
C-----------------------------------------------------------------------ADL40320
C FILE 'ENVIRL.COM'                                                     ADL40330
        COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                  ADL40340
        COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),       ADL40350
     2  NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                           ADL40360
        COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,          ADL40370
     2  KDTIME,TFACTR,IPER,PERIOD                                       ADL40380
        COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)       ADL40390
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL40400
        COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),       ADL40410
     1     PSEDCO(10)                                                   ADL40420
        COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                ADL40430
     2 SEDFL(10,10),WATINL(10)                                          ADL40440
        COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),            ADL40450
     1  WAVEL(39),TOTETA(39),WLAML(39)                                  ADL40460
        REAL INTFL,NPSFL,NPSCOL                                         ADL40470
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                     ADL40480
      LOGICAL PRINT                                                      ADL40490
      INTEGER L,E,H,B                                                    ADL40500
C----------------------------------------------------------------------- ADL40510
C----------------------------------------------------------------------- ADL40520
C THIS VERSION IS DIMENSIONED FOR 10 COMPARTMENTS AT MOST:               ADL40530
      DIMENSION Y(10),TOMBIO(2)                                          ADL40540
C WRITE PAGE HEADER:                                                     ADL40550
      IF (PRINT) WRITE(RPTOUT,100) CHEMNA,SYSTYP                         ADL40560
100   FORMAT(1H1,'  AERL-ESB MODEL OF FATE OF ORGANIC TOXICANTS',        ADL40570
     1 1X,'IN AQUATIC ECOSYSTEMS.'/1X,'CHEMICAL: ',60A1/1X,              ADL40580
     2 'ECOSYSTEM: ',60A1/1X,72(1H-))                                    ADL40590
C                                                                        ADL40600
C                                                                        ADL40610
      IF (PRINT) WRITE(RPTOUT,110)                                       ADL40620
110   FORMAT(11X,'ANALYSIS OF STEADY-STATE FATE OF ORGANIC TOXICANT')    ADL40630
C   SET UP HEADERS FOR MASS FLUX TABLE:                                  ADL40640
      IF (PRINT) WRITE(RPTOUT,600)                                       ADL40650
      IF (PRINT) WRITE(RPTOUT,120)                                       ADL40660
120   FORMAT(14X,'PROCESS',16X,'MASS FLUX',4X,                           ADL40670
     1  '% OF LOAD',4X,'HALF-LIFE*')                                     ADL40680
C----------------------------------------------------------------------- ADL40690
C ANALYSIS OF MASS FLUXES AT STEADY-STATE:                               ADL40700
C CALCULATE MASS EXPORT RATE FROM SYSTEM AT STEADY-STATE:                ADL40710
C FACTOR OF 1.E-6 CONVERTS MG/HR TO KG/HR                                ADL40720
C                                                                        ADL40730
      TOMOUT=0.                                                          ADL40740
      DO 130 J=1,KOUNT                                                   ADL40750
      TOMOUT=TOMOUT+1.E-6*Y(J)*EXPOKL(J)                                 ADL40760
130   CONTINUE                                                           ADL40770
C----------------------------------------------------------------------- ADL40780
C COMPUTE VOLATILIZATION AND TRANSFORMATIONS OF TOXICANT,                ADL40790
C ACCUMULATE TOTAL MASS FLUXES FOR ECOSYSTEM:                            ADL40800
C INITIALIZE MASS FLUXES:  ALL AS KG/HR                                  ADL40810
C VOLATILIZATION, KG/HR                                                  ADL40820
      TOMVOL=0.                                                          ADL40830
C CHEMICAL TRANSFORMATION, KG/HR                                         ADL40840
      TOMCHE=0.                                                          ADL40850
C HYDROLYSIS                                                             ADL40860
      TOMHYD=0.                                                          ADL40870
C OXIDATION                                                              ADL40880
      TOMOX=0.                                                           ADL40890
C DIRECT PHOTOLYSIS                                                      ADL40900
      TOMDP=0.                                                           ADL40910
C BIOLYSIS IN WATER COLUMN:                                              ADL40920
      TOMBIO(1)=0.                                                       ADL40930
C BIOLYSIS IN BOTTOM SEDIMENTS:                                          ADL40940
      TOMBIO(2)=0.                                                       ADL40950
C----------------------------------------------------------------------- ADL40960
C  COMPUTE PROCESS RATES FOR COMPARTMENTS IN NUMERICAL ORDER:            ADL40970
      DO 160 J=1,KOUNT                                                   ADL40980
C                                                                        ADL40990
C 1. VOLATILIZATION: ONLY FROM EPILIMNION - MG/L/HR                      ADL41000
                VOL=VOLKL(J)*Y(J)                                        ADL41010
C KG/HR TOTAL VOLATILIZATION:                                            ADL41020
                TOMVOL=TOMVOL+1.E-6*VOL*WATVOL(J)                        ADL41030
C                                                                        ADL41040
```

```
C 2. HYDROLYSIS: 3 LIMB (ACID, BASE, NEUTRAL) PROCESS, OCCURRS          ADL41050
C    IN ALL COMPARTMENTS.                                               ADL41060
                CHE=HYDRKL(J)*Y(J)                                      ADL41070
C MG/HR                                                                 ADL41080
                TOMCHE=TOMCHE+CHE*WATVOL(J)                             ADL41090
C                                                                       ADL41100
C KG/HR                                                                 ADL41110
                TOMHYD=TOMHYD+1.E-6*CHE*WATVOL(J)                       ADL41120
C                                                                       ADL41130
C 3. OXIDATION: (MG/L/HR)                                               ADL41140
C CONCENTRATION OF RO2 IS 1.E-9 M IN WATER COLUMN,                      ADL41150
C    ZERO OR THEREABOUTS IN BOTTOM SEDIMENTS.                           ADL41160
                CHE=OXIDKL(J)*Y(J)                                      ADL41170
C                                                                       ADL41180
C MG/HR                                                                 ADL41190
                TOMCHE=TOMCHE+CHE*WATVOL(J)                             ADL41200
C KG/HR                                                                 ADL41210
                TOMOX=TOMOX+1.E-6*CHE*WATVOL(J)                         ADL41220
C                                                                       ADL41230
C 4. DIRECT PHOTOLYSIS:                                                 ADL41240
                CHE=PHOTKL(J)*Y(J)                                      ADL41250
C MG/HR                                                                 ADL41260
                TOMCHE=TOMCHE+CHE*WATVOL(J)                             ADL41270
C KG/HR                                                                 ADL41280
                TOMDP=TOMDP+1.E-6*CHE*WATVOL(J)                         ADL41290
C                                                                       ADL41300
C                                                                       ADL41310
C                                                                       ADL41320
C 5. MICROBIAL TRANSFORMATION: DEPENDENT ON AQUEOUS CONCENTRATION       ADL41330
C    AND POPULATION LEVELS OF BACTERIA.                                 ADL41340
C                                                                       ADL41350
C SKIP TO BOTTOM SEDIMENTS:                                             ADL41360
      IF(INDEXS(J).EQ.1) GO TO 140                                      ADL41370
C WATER COLUMN:                                                         ADL41380
      BIO=Y(J)*BIOLKL(J)                                                ADL41390
      TOMBIO(1)=TOMBIO(1)+1.E-6*BIO*WATVOL(J)                           ADL41400
C TOMBIO CONVERTED FROM MG/L/HR TO KG/HR                                ADL41410
C                                                                       ADL41420
      GO TO 150                                                         ADL41430
140   CONTINUE                                                          ADL41440
C                                                                       ADL41450
C BOTTOM SEDIMENTS:                                                     ADL41460
      BIO=Y(J)*BIOLKL(J)                                                ADL41470
      TOMBIO(2)=TOMBIO(2)+1.E-6*BIO*WATVOL(J)                           ADL41480
C                                                                       ADL41490
150   CONTINUE                                                          ADL41500
C                                                                       ADL41510
C                                                                       ADL41520
160     CONTINUE                                                        ADL41530
C END OF TRANSFORMATION LOOP                                            ADL41540
C---------------------------------------------------------------------- ADL41550
C CONVERT MG/HR TO KG/HR:                                               ADL41560
        TOMCHE=TOMCHE*1.E-6                                             ADL41570
C                                                                       ADL41580
C COMPUTE TOTAL FLUX AND RESIDUAL ACCUMULATION RATE:                    ADL41590
        TOTLOS=TOMOUT+TOMVOL+TOMBIO(1)+TOMBIO(2)+TOMCHE                 ADL41600
        TOMACU=ABS(SYSLDL-TOTLOS)                                       ADL41610
```

```
C-----------------------------------------------------------------------ADL41620
C-----------------------------------------------------------------------ADL41630
C COMPUTE TOTAL SYSTEM-LEVEL HALF-LIFE IN HOURS:                         ADL41640
      HAFLT=0.69315*(Z(5)+Z(6))/TOTLOS                                  ADL41650
C RUN DECAY KINETIC SIMULATION FOR 2 HALF-LIVES:                        ADL41660
      TFINAL=2.*HAFLT                                                   ADL41670
C DETERMINE APPROPRIATE TEMPORAL UNIT FOR REPORTING RESULTS:            ADL41680
C 1. TIME LESS THAN 12 DAYS: REPORT IN HOURS                            ADL41690
      IF(TFINAL.LE.288.) KDTIME=1                                       ADL41700
C 2. TIME MORE THAN 12 DAYS BUT LESS THAN 1 YR:                         ADL41710
C    REPORT IN DAYS                                                     ADL41720
      IF(TFINAL.GT.288. .AND. TFINAL.LE.8766.) KDTIME=2                 ADL41730
C 3. MORE THAN ONE YEAR BUT LESS THAN 12 YEARS: REPORT IN MONTHS        ADL41740
      IF(TFINAL.GT.8766. .AND. TFINAL.LE.105192.) KDTIME=3              ADL41750
C 4. MORE THAN 12 YEARS: REPORT IN YEARS                                ADL41760
      IF(TFINAL.GT.105192.) KDTIME=4                                    ADL41770
C COMPUTE ADJUSTMENT FACTOR FOR REPORT TIMES:                           ADL41780
C 1. REPORT IN HOURS:                                                   ADL41790
      IF(KDTIME.EQ.1) TFACTR=1.0                                        ADL41800
C 2. REPORT IN DAYS:                                                    ADL41810
      IF(KDTIME.EQ.2) TFACTR=24.                                        ADL41820
C 3. REPORT IN MONTHS:                                                  ADL41830
      IF(KDTIME.EQ.3) TFACTR=730.5                                      ADL41840
C 4. REPORT IN YEARS:                                                   ADL41850
      IF(KDTIME.EQ.4) TFACTR=8766.                                      ADL41860
C CONVERT TIME FRAME TO APPROPRIATE INTEGERS:                           ADL41870
      TIME1=TFINAL/12./TFACTR                                           ADL41880
      TIME2=FLOAT(IFIX(TIME1))                                          ADL41890
      IF((TIME1-TIME2).LT.0.5) GO TO 165                                ADL41900
      TIME2=TIME2 + 1.0                                                 ADL41910
165   TINCR=TFACTR*TIME2                                                ADL41920
C TINCR RESTRICTED TO AT LEAST ONE HOUR:                                ADL41930
      IF(TINCR.LT.1.) TINCR=1.0                                         ADL41940
      TFINAL=TINCR*12.                                                  ADL41950
C TAG TIME TRACE TO BE WRITTEN TO KINE.DAT TO IDENTIFY UNITS:           ADL41960
      WRITE(KINOUT) KDTIME                                              ADL41970
C                                                                       ADL41980
C**********                                                             ADL41990
C                                                                       ADL42000
      IF (.NOT.PRINT) GO TO 215                                         ADL42010
C                                                                       ADL42020
C**********                                                             ADL42030
C COMPLETE TABLE HEADING BASED ON TIME FRAME TO BE USED:                ADL42040
C IF OUTPUT IN HOURS:                                                   ADL42050
      IF(KDTIME.EQ.1) WRITE(RPTOUT,170)                                 ADL42060
170   FORMAT(39X,'KG/HR',21X,'HOURS')                                   ADL42070
C IF OUTPUT IN DAYS:                                                    ADL42080
      IF(KDTIME.EQ.2.) WRITE(RPTOUT,180)                                ADL42090
180   FORMAT(38X,'KG/DAY',22X,'DAYS')                                   ADL42100
C IF OUTPUT IN MONTHS:                                                  ADL42110
      IF(KDTIME.EQ.3) WRITE(RPTOUT,190)                                 ADL42120
190   FORMAT(38X,'KG/MONTH',19X,'MONTHS')                               ADL42130
C IF OUTPUT IN YEARS:                                                   ADL42140
      IF(KDTIME.EQ.4) WRITE(RPTOUT,200)                                 ADL42150
200   FORMAT(39X,'KG/YR',21X,'YEARS')                                   ADL42160
      WRITE(RPTOUT,210)                                                 ADL42170
210   FORMAT(7X,24(1H-),6X,9(1H-),2(4X,9(1H-)))                         ADL42180
```

```
C**********                                                             ADL42190
215   CONTINUE                                                          ADL42200
C**********                                                             ADL42210
C                                                                       ADL42220
C  CONVERT MASS FLUXES AND LOAD TO KG/(APPROPRIATE TIME UNIT):          ADL42230
C  IF WORKING IN HOURS, SIMPLY GO ON:                                   ADL42240
       IF(KDTIME.EQ.1) GO TO 220                                        ADL42250
C                                                                       ADL42260
C CONVERT MASS FLUXES BY TIME FACTOR:                                   ADL42270
        SYSLDL=SYSLDL*TFACTR                                            ADL42280
        TOMHYD=TOMHYD*TFACTR                                            ADL42290
        TOMOX=TOMOX*TFACTR                                              ADL42300
        TOMDP=TOMDP*TFACTR                                              ADL42310
        TOMSP=TOMSP*TFACTR                                              ADL42320
        TOMBIO(1)=TOMBIO(1)*TFACTR                                      ADL42330
        TOMBIO(2)=TOMBIO(2)*TFACTR                                      ADL42340
        TOMVOL=TOMVOL*TFACTR                                            ADL42350
        TOMOUT=TOMOUT*TFACTR                                            ADL42360
        TOMCHE=TOMCHE*TFACTR                                            ADL42370
        TOMACU=TOMACU*TFACTR                                            ADL42380
C                                                                       ADL42390
220     CONTINUE                                                        ADL42400
C-----------------------------------------------------------------------ADL42410
C-----------------------------------------------------------------------ADL42420
C WRITE OUT PROCESS MASS FLUXES:                                        ADL42430
C------------------------------                                         ADL42440
C  CHEMICAL TRANFORMATIONS:                                             ADL42450
C-----------------------------------------------------------------------ADL42460
C       HYDROLYSIS:                                                     ADL42470
C       -----------                                                     ADL42480
        PERCET=100.*TOMHYD/SYSLDL                                       ADL42490
         IF(TOMHYD.NE.0) GO TO 240                                      ADL42500
C NO HYDROLYSIS                                                         ADL42510
         IF (PRINT) WRITE(RPTOUT,230) TOMHYD,PERCET                     ADL42520
        GO TO 260                                                       ADL42530
230      FORMAT(7X,'HYDROLYSIS',19X,G11.4,5X,F5.2,9X,3(1H-))            ADL42540
240     CONTINUE                                                        ADL42550
         HAFLT=0.69315*(Z(5)+Z(6))/TOMHYD                               ADL42560
        IF (PRINT) WRITE(RPTOUT,250) TOMHYD,PERCET,HAFLT                ADL42570
250     FORMAT(7X,'HYDROLYSIS',19X,1PG11.4,4X,0PF6.2,5X,1PG11.4)        ADL42580
260     CONTINUE                                                        ADL42590
C-----------------------------------------------------------------------ADL42600
C        OXIDATION:                                                     ADL42610
C        ----------                                                     ADL42620
        PERCET=100.*TOMOX/SYSLDL                                        ADL42630
         IF(TOMOX.NE.0.) GO TO 280                                      ADL42640
C NO OXIDATION:                                                         ADL42650
          IF (PRINT) WRITE(RPTOUT,270) TOMOX,PERCET                     ADL42660
          GO TO 300                                                     ADL42670
270      FORMAT(7X,'OXIDATION',20X,G11.4,5X,F5.2,9X,3(1H-))             ADL42680
C                                                                       ADL42690
280     CONTINUE                                                        ADL42700
          HAFLT=0.69315*(Z(5)+Z(6))/TOMOX                               ADL42710
          IF (PRINT) WRITE(RPTOUT,290) TOMOX,PERCET,HAFLT               ADL42720
290      FORMAT(7X,'OXIDATION',20X,1PG11.4,4X,0PF6.2,5X,1PG11.4)        ADL42730
300     CONTINUE                                                        ADL42740
C-----------------------------------------------------------------------ADL42750
```

```
C          PHOTOLYSIS:
C          -----------
        TOMPHO = TOMDP
        PERCET=100.*TOMPHO/SYSLDL
           IF(TOMPHO.NE.0.) GO TO 320
C NO PHOTOLYSIS:
           IF (PRINT) WRITE(RPTOUT,310) TOMPHO,PERCET
          GO TO 340
310     FORMAT(7X,'PHOTOLYSIS',19X,G11.4,5X,F5.2,9X,3(1H-))
C
320     CONTINUE
           HAFLT=0.69315*(Z(5)+Z(6))/TOMPHO
           IF (PRINT) WRITE(RPTOUT,330) TOMPHO,PERCET,HAFLT
330     FORMAT(7X,'PHOTOLYSIS',19X,1PG11.4,4X,0PF6.2,5X,1PG11.4)
340     CONTINUE
C-----------------------------------------------------------------------
C         TOTAL CHEMICAL TRANSFORMATIONS:
C         ------------------------------
        CHEMPC=100.*TOMCHE/SYSLDL
           IF(TOMCHE.NE.0.) GO TO 360
C NO CHEMICAL TRANSFORMATIONS:
           IF (PRINT) WRITE(RPTOUT,350) TOMCHE,CHEMPC
350     FORMAT(3X,'ALL CHEMICAL PROCESSES',11X,G11.4,5X,F5.2,9X,3(1H-))
           GO TO 380
C
360     CONTINUE
           HAFLT=0.69315*(Z(5)+Z(6))/TOMCHE
           IF (PRINT) WRITE(RPTOUT,370) TOMCHE,CHEMPC,HAFLT
370     FORMAT(3X,'ALL CHEMICAL PROCESSES',11X,1PG11.4,4X,0PF6.2,5X,
     2  1PG11.4)
C-----------------------------------------------------------------------
C       MICROBIAL TRANFORMATIONS:
C       ------------------------
380     IF (PRINT) WRITE(RPTOUT,600)
C       WATER COLUMN:
C       ------------
        PERCET=100.*TOMBIO(1)/SYSLDL
        IF(TOMBIO(1).NE.0.) GO TO 400
           IF (PRINT) WRITE(RPTOUT,390) TOMBIO(1),PERCET
390     FORMAT(7X,'WATER COLUMN (BACTERIA)',6X,G11.4,5X,F5.2,9X,3(1H-))
           GO TO 420
C
400     CONTINUE
           HAFLW=0.69315*Z(5)/TOMBIO(1)
           IF (PRINT) WRITE(RPTOUT,410) TOMBIO(1),PERCET,HAFLW
410     FORMAT(7X,'WATER COLUMN (BACTERIA)',6X,1PG11.4,4X,0PF6.2,5X,
     2 1PG11.4)
C       BOTTOM SEDIMENTS:
C       ----------------
        PERCET=100.*TOMBIO(2)/SYSLDL
420     IF(TOMBIO(2).NE.0.) GO TO 440
         IF (PRINT) WRITE(RPTOUT,430) TOMBIO(2),PERCET
430     FORMAT(7X,'BOTTOM SEDIMENTS (BACTERIA)',2X,G11.4,5X,
     1 F5.2,9X,3(1H-))
          GO TO 460
440     CONTINUE
        HAFLS=0.69315*Z(6)/TOMBIO(2)
```

```
       IF (PRINT) WRITE(RPTOUT,450) TOMBIO(2),PERCET,HAFLS                ADL43330
450    FORMAT(7X,'BOTTOM SEDIMENTS (BACTERIA)',2X,1PG11.4,4X,              ADL43340
     1 0PF6.2,5X,1PG11.4)                                                 ADL43350
C-----------------------------------------------------------------------  ADL43360
C      TOTAL MICROBIAL TRANSFORMATION:                                    ADL43370
C      ------------------------------                                     ADL43380
C (UNDER THE FIRST-ORDER ASSUMPTIONS USED IN THIS ANALYSIS,               ADL43390
C  THE DERIVATIVES CAN BE SUMMED AND RELATED TO TOTAL MASS.)              ADL43400
C                                                                         ADL43410
460    TOTBIO=TOMBIO(1)+TOMBIO(2)                                         ADL43420
       BIOPCT=100.*TOTBIO/SYSLDL                                          ADL43430
        IF(TOTBIO.NE.0.) GO TO 480                                        ADL43440
C NO BIOLOGICAL TRANSFORMATIONS:                                          ADL43450
       IF (PRINT) WRITE(RPTOUT,470) TOTBIO,BIOPCT                         ADL43460
470    FORMAT(3X,'TOTAL BIOLYSIS',19X,G11.4,5X,F5.2,9X,3(1H-))            ADL43470
C                                                                         ADL43480
       GO TO 500                                                          ADL43490
C                                                                         ADL43500
C                                                                         ADL43510
480    CONTINUE                                                           ADL43520
       HAFLT=0.69315*(Z(5)+Z(6))/TOTBIO                                   ADL43530
       IF (PRINT) WRITE(RPTOUT,490) TOTBIO,BIOPCT,HAFLT                   ADL43540
490    FORMAT(3X,'TOTAL BIOLYSIS',19X,1PG11.4,4X,0PF6.2,5X,1PG11.4)       ADL43550
C                                                                         ADL43560
500    CONTINUE                                                           ADL43570
       IF (PRINT) WRITE(RPTOUT,600)                                       ADL43580
C-----------------------------------------------------------------------  ADL43590
C  VOLATILIZATION AND WATER-BORNE EXPORT:                                 ADL43600
C  -------------------------------------                                  ADL43610
C      VOLATILIZATION:                                                    ADL43620
C      ---------------                                                    ADL43630
       VOLPCT=100.*TOMVOL/SYSLDL                                          ADL43640
       IF(TOMVOL.NE.0.) GO TO 520                                         ADL43650
       IF (PRINT) WRITE(RPTOUT,510) TOMVOL,VOLPCT                         ADL43660
510    FORMAT(7X,'VOLATILIZATION',15X,G11.4,5X,F5.2,9X,3(1H-))            ADL43670
       GO TO 540                                                          ADL43680
C                                                                         ADL43690
520    CONTINUE                                                           ADL43700
       HAFLT=0.69315*(Z(5)+Z(6))/TOMVOL                                   ADL43710
       IF (PRINT) WRITE(RPTOUT,530) TOMVOL,VOLPCT,HAFLT                   ADL43720
530    FORMAT(7X,'VOLATILIZATION',15X,1PG11.4,4X,0PF6.2,5X,1PG11.4)       ADL43730
C                                                                         ADL43740
540    CONTINUE                                                           ADL43750
C-----------------------------------------------------------------------  ADL43760
C      WATER-BORNE EXPORT:                                                ADL43770
C      -------------------                                                ADL43780
       EXPPCT=100.*TOMOUT/SYSLDL                                          ADL43790
       IF(TOMOUT.EQ.0.) GO TO 556                                         ADL43800
       HAFLT=0.69315*(Z(5)+Z(6))/TOMOUT                                   ADL43810
       IF (PRINT) WRITE(RPTOUT,550) TOMOUT,EXPPCT,HAFLT                   ADL43820
550  FORMAT(7X,'WATER-BORNE EXPORT',11X,1PG11.4,4X,0PF6.2,5X,1PG11.4)     ADL43830
C                                                                         ADL43840
       GO TO 559                                                          ADL43850
556    CONTINUE                                                           ADL43860
C NO WATER-BORNE EXPORT:                                                  ADL43870
       IF (PRINT) WRITE(RPTOUT,557) TOMOUT,EXPPCT                         ADL43880
557    FORMAT(7X,'WATER-BORNE EXPORT',11X,G11.4,5X,F5.2,9X,3(1H-))        ADL43890
```

```
559     CONTINUE
        IF (PRINT) WRITE(RPTOUT,600)
C-----------------------------------------------------------------------
        TOTLOS=TOMOUT+TOMVOL+TOTBIO+TOMCHE
C LOSS BY ALL PROCESSES COMBINED:
        PERCET=100.*TOTLOS/SYSLDL
        HAFLT=0.69315*(Z(5)+Z(6))/TOTLOS
        IF (PRINT) WRITE(RPTOUT,560) TOTLOS,PERCET,HAFLT
560     FORMAT(1X,'TRANSFORMATION AND TRANSPORT',7X,1PG11.4,
     1  4X,0PF6.2,5X,1PG11.4)
        IF (PRINT) WRITE(RPTOUT,570) SYSLDL
570     FORMAT(1X,'TOTAL SYSTEM LOAD',18X,1PG11.4)
C-----------------------------------------------------------------------
C       REMAINING ACCUMULATION RATE:
C       --------------------------
         PERCET=100.*TOMACU/SYSLDL
         IF (PRINT) WRITE(RPTOUT,580) TOMACU,PERCET
580     FORMAT(1X,'RESIDUAL ACCUMULATION RATE:',8X,1PG11.4,5X,0PF5.2)
C
        IF (PRINT) WRITE(RPTOUT,600)
        IF (PRINT) WRITE(RPTOUT,590)
590     FORMAT(1X,'* HALF-LIVES ARE ESTIMATES BASED ON A FIRST-ORDER ',
     1  'RATE APPROXIMATION.')
600     FORMAT(1X,72(1H-))
C-----------------------------------------------------------------------
C COMPLETION OF ANALYSIS OF STEADY-STATE
C-----------------------------------------------------------------------
        RETURN
        END
C FILE 'FCT.FTN'
        SUBROUTINE FCT(TIME,Y,DY)
C THIS SUBROUTINE PERFORMS CALCULATIONS FOR THE RATES OF
C TRANSPORT AND DEGRADATION OF TOXICANT IN THE ECOSYSTEM.
C-----------------------------------------------------------------------
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG
     1,NPSEDG,INTFLG
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT
      LOGICAL*1 CHEMNA,SYSTYP
      COMMON/NAME1G/CHEMNA(60)
      COMMON/CONTRG/SPFLG(5)
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)
     1,KBHG(3,5),EBHG(3,5)
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)          ADL44470
     1,IFLLDG(10)                                                        ADL44480
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                 ADL44490
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)    ADL44500
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                   ADL44510
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)     ADL44520
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)      ADL44530
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)               ADL44540
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG          ADL44550
     1,DFACG(10)                                                         ADL44560
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                         ADL44570
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                        ADL44580
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)           ADL44590
     1,INTFLG(10)                                                        ADL44600
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT         ADL44610
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                       ADL44620
C-----------------------------------------------------------------       ADL44630
C FILE 'CHEML.COM'                                                       ADL44640
      COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)             ADL44650
      COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),         ADL44660
    2 VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                       ADL44670
      COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                          ADL44680
      COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                              ADL44690
      COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),              ADL44700
    2 CHEMPC,BIOPCT,VOLPCT,EXPPCT                                        ADL44710
      COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)                  ADL44720
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                          ADL44730
C-----------------------------------------------------------------       ADL44740
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                    ADL44750
C-----------------------------------------------------------------       ADL44760
C FILE 'ENVIRL.COM'                                                      ADL44770
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                     ADL44780
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),          ADL44790
    2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                              ADL44800
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,             ADL44810
    2 KDTIME,TFACTR,IPER,PERIOD                                          ADL44820
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)          ADL44830
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                        ADL44840
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),          ADL44850
    1     PSEDCO(10)                                                     ADL44860
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                   ADL44870
    2 SEDFL(10,10),WATINL(10)                                            ADL44880
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),               ADL44890
    1 WAVEL(39),TOTETA(39),WLAML(39)                                     ADL44900
      REAL INTFL,NPSFL,NPSCOL                                            ADL44910
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                     ADL44920
      LOGICAL PRINT                                                      ADL44930
      INTEGER L,E,H,B                                                    ADL44940
C-----------------------------------------------------------------       ADL44950
C-----------------------------------------------------------------       ADL44960
C THIS VERSION DIMENSIONED FOR 10 COMPARTMENTS AT MOST:                  ADL44970
      DIMENSION Y(10),DY(10)                                             ADL44980
C Y IS CONSTITUENT CONCENTRATION (MG/L) REFERRED TO                      ADL44990
C THE AQUEOUS PHASE OF COMPARTMENT.                                      ADL45000
C-----------------------------------------------------------------       ADL45010
C CONDENSED EQUATIONS FOR TIME-TRACE COMPUTATIONS:                       ADL45020
    DO 110 J=1,KOUNT                                                     ADL45030
```

```
C COMPUTE PART OF DERIVATIVE DUE TO EXOGENENOUS LOADS AND FIRST-ORDER
C TERMS:
         DY(J)= -Y(J)*TOTKL(J)+CONLDL(J)
C ADD IN INTERNAL (RECYCLE) LOADINGS:
         DO 100 I=1,KOUNT
          DY(J)=DY(J)+Y(I)*INTINL(I,J)
100      CONTINUE
110      CONTINUE
C-----------------------------------------------------------------------
         RETURN
         END
C FILE 'OUTP.FTN'
      SUBROUTINE OUTP(Y,W)
        DIMENSION Y(1),W(1)
C-----------------------------------------------------------------------
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:
       REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG
     1,NPSEDG,INTFLG
      INTEGER   SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT
      LOGICAL*1 CHEMNA,SYSTYP
      COMMON/NAME1G/CHEMNA(60)
      COMMON/CONTRG/SPFLG(5)
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)
     1,KBHG(3,5),EBHG(3,5)
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)
     1,IFLLDG(10)
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG
     1,DFACG(10)
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)
     1,INTFLG(10)
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:
C-----------------------------------------------------------------------
C FILE 'CHEML.COM'
      COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)
      COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),        ADL45610
     2 VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                     ADL45620
      COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                         ADL45630
      COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                             ADL45640
      COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),             ADL45650
     2 CHEMPC,BIOPCT,VOLPCT,EXPPCT                                      ADL45660
      COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)                 ADL45670
      REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                         ADL45680
C-----------------------------------------------------------------------ADL45690
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                   ADL45700
C-----------------------------------------------------------------------ADL45710
C FILE 'ENVIRL.COM'                                                     ADL45720
      COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                    ADL45730
      COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),         ADL45740
     2 NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                            ADL45750
      COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,            ADL45760
     2 KDTIME,TFACTR,IPER,PERIOD                                        ADL45770
      COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)         ADL45780
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL45790
      COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),         ADL45800
     1     PSEDCO(10)                                                   ADL45810
      COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                  ADL45820
     2 SEDFL(10,10),WATINL(10)                                          ADL45830
      COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),              ADL45840
     1 WAVEL(39),TOTETA(39),WLAML(39)                                   ADL45850
      REAL INTFL,NPSFL,NPSCOL                                           ADL45860
      REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                    ADL45870
      LOGICAL PRINT                                                     ADL45880
      INTEGER L,E,H,B                                                   ADL45890
C-----------------------------------------------------------------------ADL45900
C-----------------------------------------------------------------------ADL45910
C CALCULATE CONCENTRATIONS AND TOTAL ECOSYSTEM POLLUTANT MASSES:        ADL45920
C  ZERO 'Z' VALUES:                                                     ADL45930
      DO 100 I=1,18                                                     ADL45940
            Z(I)=0.                                                     ADL45950
100   CONTINUE                                                          ADL45960
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ADL45970
C  COMPUTE MEAN CONCENTRATIONS:                                         ADL45980
C DO COMPARTMENTS IN NUMERICAL ORDER:                                   ADL45990
      DO 120 J=1,KOUNT                                                  ADL46000
C SKIP TO BOTTOM SEDIMENT:                                              ADL46010
        IF(INDEXS(J).EQ.1) GO TO 110                                    ADL46020
C WATER COLUMN COMPARTMENTS:                                            ADL46030
C DISSOLVED:                                                            ADL46040
      Z(1)=Z(1)+ALPHA(16,J)*Y(J)                                        ADL46050
C SORBED MG/KG:                                                         ADL46060
      Z(2)=Z(2)+ALPHA(17,J)*Y(J)/SEDCOL(J)                              ADL46070
C MASS, KG (1.E-6 KG/MG)                                                ADL46080
      Z(5)=Z(5)+Y(J)*WATVOL(J)*1.E-6                                    ADL46090
      GO TO 120                                                         ADL46100
C BOTTOM SEDIMENTS:                                                     ADL46110
110   CONTINUE                                                          ADL46120
C MG/L DISSOLVED IN PORE WATER:                                         ADL46130
      Z(3)=Z(3)+ALPHA(16,J)*Y(J)                                        ADL46140
C SORBED MG/KG                                                          ADL46150
      Z(4)=Z(4)+ALPHA(17,J)*Y(J)/SEDCOL(J)                              ADL46160
C POLLUTANT MASS IN KG (1.E-6 KG/MG)                                    ADL46170
```

```
      Z(6)=Z(6)+Y(J)*WATVOL(J)*1.E-6                                    ADL46180
120   CONTINUE                                                          ADL46190
C COMPUTE AVERAGES IN WATER COLUMN:                                     ADL46200
C MG/L  DISSOLVED                                                       ADL46210
      Z(1)=Z(1)/KOUNTW                                                  ADL46220
C MG/KG SORBED:                                                         ADL46230
      Z(2)=Z(2)/KOUNTW                                                  ADL46240
C COMPUTE AVERAGES IN BOTTOM SEDIMENTS:                                 ADL46250
C MG/L DISSOLVED:                                                       ADL46260
      Z(3)=Z(3)/KOUNTS                                                  ADL46270
C MG/KG SORBED                                                          ADL46280
      Z(4)=Z(4)/KOUNTS                                                  ADL46290
C---------------------------------------------------------------------- ADL46300
      TTEMP=T/TFACTR                                                    ADL46310
C REPORT TIME COURSE IN RUN LOG:                                        ADL46320
      IF (PRINT) WRITE(RPTOUT,130) TTEMP,(Z(J),J=1,6)                   ADL46330
130   FORMAT(1X,F5.0,4(1X,1PG10.3),1X,G11.4,1X,G10.3)                   ADL46340
C PUT THIS DATA IN PLOT FILE FOR KINETICS:                              ADL46350
      WRITE(KINOUT) TTEMP,(Z(J),J=1,6)                                  ADL46360
C WRITE COMPLETE STATE VARIABLE VECTOR TO PLOTTING FILE:                ADL46370
      DO 140 J=1,KOUNT                                                  ADL46380
C FIRST ELEMENT IN KINETICS PLOT FILE (KINE.DAT) IS TOTAL CONC.         ADL46390
C (MG/L IN WATER COLUMN COMPARTMENTS, MG/KG IN BOTTOM SEDIMENTS)        ADL46400
      Z(7)=Y(J)                                                         ADL46410
      IF(INDEXS(J).EQ.1) Z(7)=Y(J)/SEDCOL(J)                            ADL46420
C NEXT ELEMENT IN PLOT FILE IS FREE (DISSOLVED) TOXICANT                ADL46430
C IN WATER PHASE (MG/L).                                                ADL46440
        Z(8)=ALPHA(16,J)*Y(J)                                           ADL46450
C THEN TOXICANT SORBED IN/ON SOLID SEDIMENTS: (MG/KG SOLIDS)            ADL46460
        Z(9)=ALPHA(17,J)*Y(J)/SEDCOL(J)                                 ADL46470
C NEXT IS TOXICANT SORBED IN/ON BIOTA (BACTERIA) IN UG/G:               ADL46480
        Z(10)=ALPHA(18,J)*Y(J)/BIOTOL(J)                                ADL46490
C FINALLY, COMPARTMENT MASS IN GRAMS/SQUARE METER:                      ADL46500
C 1.E-3 FACTOR CONVERTS MG TO GRAMS                                     ADL46510
        Z(11)=1.E-3*Y(J)*WATVOL(J)/AREAG(J)                             ADL46520
C WRITE THESE QUANTITIES TO KINETICS FILE (KINE.DAT):                   ADL46530
      WRITE(KINOUT) TTEMP,J,(Z(I),I=7,11)                               ADL46540
140   CONTINUE                                                          ADL46550
      RETURN                                                            ADL46560
      END                                                               ADL46570
C FILE 'SUMUP.FTN'                                                      ADL46580
      SUBROUTINE SUMUP                                                  ADL46590
C---------------------------------------------------------------------- ADL46600
C SUMUP. SUMUP WRITES A SINGLE-PAGE SUMMARY OF THE PRIOR ANALYSES TO    ADL46610
C FILE 'REPORT.DAT', ALONG WITH AN ESTIMATE OF THE ECOSYSTEM SELF-      ADL46620
C CLEANSING TIME DERIVED FROM AN ANALYSIS OF THE KINETIC DECAY DATA     ADL46630
C COMPUTED BY THE INTEGRATION PACKAGE (RKF45).                          ADL46640
C-----------------------------------------------------------------------ADL46650
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL46660
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL46670
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL46680
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL46690
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG     ADL46700
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL46710
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL46720
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL46730
     1,NPSEDG,INTFLG                                                    ADL46740
```

```
      INTEGER    SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT
      LOGICAL*1 CHEMNA,SYSTYP
      COMMON/NAME1G/CHEMNA(60)
      COMMON/CONTRG/SPFLG(5)
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)
     1,KBHG(3,5),EBHG(3,5)
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)
     1,IFLLDG(10)
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG
     1,DFACG(10)
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)
     1,INTFLG(10)
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:
C-----------------------------------------------------------------------
C FILE 'CHEML.COM'
         COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)
         COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),
     2   VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)
         COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)
         COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV
         COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),
     2   CHEMPC,BIOPCT,VOLPCT,EXPPCT
         COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)
         REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L
C-----------------------------------------------------------------------
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:
C-----------------------------------------------------------------------
C FILE 'ENVIRL.COM'
         COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)
         COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),
     2   NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL
         COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,
     2   KDTIME,TFACTR,IPER,PERIOD
         COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)
     1,JSAV1,JSAV2,ICALL,NPX,NCON
         COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),
     1      PSEDCO(10)
         COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),
     2 SEDFL(10,10),WATINL(10)
         COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),
     1  WAVEL(39),TOTETA(39),WLAML(39)                                 ADL47320
C**********                                                            ADL47330
      LOGICAL NPS,DOT                                                  ADL47340
      COMMON/NPSBLK/NPS,LUNPS,LUNPS2,NPSPER                            ADL47350
      COMMON/DOTBLK/DOT,LUDOT                                          ADL47360
C**********                                                            ADL47370
       REAL INTFL,NPSFL,NPSCOL                                         ADL47380
       REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                  ADL47390
       LOGICAL PRINT                                                   ADL47400
       INTEGER L,E,H,B                                                 ADL47410
C--------------------------------------------------------------------- ADL47420
C--------------------------------------------------------------------- ADL47430
C SUMMARIZE RESULTS OF RUN:                                            ADL47440
C TABLE TITLE:                                                         ADL47450
       IF (PRINT) WRITE(RPTOUT,100)                                    ADL47460
100    FORMAT(23X,'EXPOSURE ANALYSIS SUMMARY'/1X,72(1H-))              ADL47470
C                                                                      ADL47480
C ---------------------------------------------------------------------ADL47490
C COMPUTE INITIAL MASS DISTRIBUTION:                                   ADL47500
       PCTWAT=100.*QWSAV/QTSAV                                         ADL47510
       PCTSED=100.*QSSAV/QTSAV                                         ADL47520
C----------------------------------------------------------------------ADL47530
C                                                                      ADL47540
C REPORT SUMMARY:                                                      ADL47550
       IF (PRINT) WRITE(RPTOUT,110) DOMAX(3),DOMAX(2),DOMAX(8)         ADL47560
110    FORMAT(1X,'EXPOSURE:'/2X,'A. MAXIMUM CONC. IN WATER COLUMN:',   ADL47570
     11PG9.2,' MG/L DISSOLVED,',G9.2,' TOT'/5X,'MAX. CONC. IN BOTTOM', ADL47580
     2' SEDIMENT:',G9.2,' MG/L DISSOLVED IN PORE WATER')               ADL47590
C                                                                      ADL47600
       IF (PRINT) WRITE(RPTOUT,120) DOMAX(5),DOMAX(10),DOMAX(7)        ADL47610
120    FORMAT(2X,'B. BIOSORPTION - MAX. CONCENTRATION - PLANKTON:',    ADL47620
     11PG9.2,' UG/G'/40X,'BENTHOS:',1X,G9.2,' UG/G'/2X,                ADL47630
     2'C. MAXIMUM TOT. CONC. IN SEDIMENT DEPOSITS:',G9.2,' MG/KG (DRY', ADL47640
     3' WEIGHT)')                                                      ADL47650
       IF (PRINT) WRITE(RPTOUT,130) QTSAV,PCTWAT,PCTSED                ADL47660
130    FORMAT(1X,'FATE:'/2X,'A. TOTAL STEADY-STATE ACCUMULATION:',     ADL47670
     11PG9.2,' KG;',0PF6.2,'% IN WATER COL.,'/7X,F6.2,                 ADL47680
     2'% IN BOTTOM SEDIMENTS.')                                        ADL47690
C                                                                      ADL47700
C**********                                                            ADL47710
      IF ((.NOT.PRINT).OR.DOT.OR.NPS) RETURN                           ADL47720
C**********                                                            ADL47730
       IF(KDTIME.EQ.1) WRITE(RPTOUT,140) SYSLDL,CHEMPC                 ADL47740
       IF(KDTIME.EQ.2) WRITE(RPTOUT,150) SYSLDL,CHEMPC                 ADL47750
       IF(KDTIME.EQ.3) WRITE(RPTOUT,160) SYSLDL,CHEMPC                 ADL47760
       IF(KDTIME.EQ.4) WRITE(RPTOUT,170) SYSLDL,CHEMPC                 ADL47770
140    FORMAT(2X,'B. TOTAL LOAD:',1PG9.2,' KG/HOUR - DISPOSITION:',    ADL47780
     10PF7.2,'% VIA CHEMICAL')                                         ADL47790
150    FORMAT(2X,'B. TOTAL LOAD:',1PG9.2,' KG/DAY - DISPOSITION:',     ADL47800
     10PF7.2,'% VIA CHEMICAL')                                         ADL47810
160    FORMAT(2X,'B. TOTAL LOAD:',1PG9.2,' KG/MONTH - DISPOSITION:',   ADL47820
     10PF7.2,'% VIA CHEMICAL')                                         ADL47830
170    FORMAT(2X,'B. TOTAL LOAD:',1PG9.2,' KG/YEAR - DISPOSITION:',    ADL47840
     10PF7.2,'% VIA CHEMICAL')                                         ADL47850
C                                                                      ADL47860
      WRITE(RPTOUT,180) BIOPCT,VOLPCT,EXPPCT                           ADL47870
180    FORMAT(7X,'TRANSFORMATIONS,',F7.2,'% BIOTRANSFORMED,',F7.2,     ADL47880
     1'% VOLATILIZED,'/6X,F7.2,'% EXPORTED VIA OTHER PATHWAYS.')        ADL47890
C------------------------------------------------------------------------ ADL47900
C**********                                                              ADL47910
185     CONTINUE                                                          ADL47920
C**********                                                              ADL47930
C ESTIMATE PERSISTENCE FROM RESULTS OF DECAY SIMULATION:                  ADL47940
C (SKIP THIS IF INTEGRATOR FAILED:)                                       ADL47950
        IF(IFLAG.GE.8) GO TO 280                                          ADL47960
        HAFLW=0.                                                          ADL47970
        HAFLS=0.                                                          ADL47980
C IF SIMULATION BADLY OVERRUNS TRUE HALFLIVES, FINAL MASSES               ADL47990
C MAY BE ZERO OR EVEN SLIGHTLY NEGATIVE.  HENCE:                          ADL48000
        IF(Z(5).LT.0.) Z(5)=0.                                            ADL48010
        IF(Z(6).LT.0.) Z(6)=0.                                            ADL48020
C COMPUTE EFFECTIVE FIRST-ORDER RATE COEFFICIENTS:                        ADL48030
C WATER COLUMN:                                                           ADL48040
        IF(Z(5).GT.0.) AKW=TFACTR*(-ALOG(Z(5)/QWSAV))/TFINAL              ADL48050
C BOTTOM SEDIMENTS:                                                       ADL48060
        IF(Z(6).GT.0.) AKS=TFACTR*(-ALOG(Z(6)/QSSAV))/TFINAL              ADL48070
C ESTIMATE FIRST-ORDER HALF LIFE FROM FULL SYSTEM SIMULATION:             ADL48080
C WATER COLUMN:                                                           ADL48090
        IF (AKW.EQ.0.) HAFLW=999999.                                      ADL48100
        IF((Z(5).GT.0.) .AND. (AKW.NE.0.)) HAFLW=0.69315/AKW              ADL48110
C BOTTOM SEDIMENTS:                                                       ADL48120
        IF (AKS.EQ.0.) HAFLS=999999.                                      ADL48130
        IF((Z(6).GT.0.) .AND. (AKS.NE.0.)) HAFLS=0.69315/AKS              ADL48140
C -----------------------------------------------------------------------ADL48150
C CONVERT ENDPOINT ON DECAY SIMULATION TO REPORT UNITS:                   ADL48160
        TIMEPR=TFINAL/TFACTR                                              ADL48170
        IF(KDTIME.EQ.1) WRITE(RPTOUT,190) TIMEPR                          ADL48180
        IF(KDTIME.EQ.2) WRITE(RPTOUT,200) TIMEPR                          ADL48190
        IF(KDTIME.EQ.3) WRITE(RPTOUT,210) TIMEPR                          ADL48200
        IF(KDTIME.EQ.4) WRITE(RPTOUT,220) TIMEPR                          ADL48210
190     FORMAT(1X,'PERSISTENCE:',/2X,'A. AT THE END OF A',1PG10.3,        ADL48220
     1 ' HOUR RECOVERY PERIOD, THE WATER COLUMN HAD')                     ADL48230
200     FORMAT(1X,'PERSISTENCE:',/2X,'A. AT THE END OF A',1PG10.3,        ADL48240
     1 ' DAY RECOVERY PERIOD, THE WATER COLUMN HAD')                      ADL48250
210     FORMAT(1X,'PERSISTENCE:',/2X,'A. AT THE END OF A',1PG10.3,        ADL48260
     1 ' MONTH RECOVERY PERIOD, THE WATER COLUMN HAD')                    ADL48270
220     FORMAT(1X,'PERSISTENCE:',/2X,'A. AT THE END OF A',1PG10.3,        ADL48280
     1 ' YEAR RECOVERY PERIOD, THE WATER COLUMN HAD')                     ADL48290
C------------------------------------------------------------------------ ADL48300
C TIME TO REMOVE CA. 95% OF STEADY-STATE RESIDENT TOXICANT:               ADL48310
C (ESTIMATED AS WEIGHTED AVERAGE OF FIRST-ORDER APPROXIMATIONS OF         ADL48320
C OVERALL WATER COLUMN AND BOTTOM SEDIMENT HALF-LIVES.)                   ADL48330
        DECTIM=5.*((PCTWAT/100.)*HAFLW + (PCTSED/100.)*HAFLS)             ADL48340
C------------------------------------------------------------------------ADL48350
C COMPUTE PERCENT REMAINING AT END OF DECAY SIMULATION:                   ADL48360
        PCTWAT=100.*(1.-(Z(5)/QWSAV))                                     ADL48370
        PCTSED=0.                                                         ADL48380
C ALLOW FOR RUN WITHOUT SEDIMENT COMPARTMENTS:                            ADL48390
        IF(QSSAV.NE.0.) PCTSED=100.*(1.-(Z(6)/QSSAV))                     ADL48400
        PCTTOT=100.*(1.-((Z(5)+Z(6))/QTSAV))                              ADL48410
        WRITE(RPTOUT,230) PCTWAT,PCTSED,PCTTOT                            ADL48420
230     FORMAT(7X,'LOST',F7.2,'% OF ITS INITIAL TOXICANT BURDEN; THE',    ADL48430
     1' SEDIMENTS HAD'/7X,'LOST',F7.2,'% OF THEIR INITIAL BURDEN (',      ADL48440
     2 F7.2,'% REMOVAL OVERALL).')                                        ADL48450
```

```
C---------------------------------------------------------------------- ADL48460
      IF(KDTIME.EQ.1) WRITE(RPTOUT,240) DECTIM                         ADL48470
      IF(KDTIME.EQ.2) WRITE(RPTOUT,250) DECTIM                         ADL48480
      IF(KDTIME.EQ.3) WRITE(RPTOUT,260) DECTIM                         ADL48490
      IF(KDTIME.EQ.4) WRITE(RPTOUT,270) DECTIM                         ADL48500
240   FORMAT(2X,'B. SYSTEM SELF-PURIFICATION TIME IS ROUGHLY',1PG11.4,ADL48510
     1' HOURS.')                                                        ADL48520
250   FORMAT(2X,'B. SYSTEM SELF-PURIFICATION TIME IS ROUGHLY',1PG11.4,ADL48530
     1' DAYS.')                                                         ADL48540
260   FORMAT(2X,'B. SYSTEM SELF-PURIFICATION TIME IS ROUGHLY',1PG11.4,ADL48550
     1' MONTHS.')                                                       ADL48560
270   FORMAT(2X,'B. SYSTEM SELF-PURIFICATION TIME IS ROUGHLY',1PG11.4,ADL48570
     1' YEARS.')                                                        ADL48580
C---------------------------------------------------------------------- ADL48590
280   CONTINUE                                                          ADL48600
      RETURN                                                            ADL48610
      END                                                               ADL48620
C FILE 'RKF45.FTN'                                                      ADL48630
      SUBROUTINE RKF45(Y,FCT,OUTP)                                      ADL48640
C-----------------------------------------------------------------------ADL48650
C RKF45. RKF45 IS A 4TH-5TH ORDER RUNGE-KUTTA INTEGRATION ROUTINE.  THE ADL48660
C INTEGRATION IS CARRIED OUT OVER 2 (ESTIMATED) HALF-LIVES, USING THE   ADL48670
C STEADY-STATE CONCENTRATIONS AS INITIAL CONDITIONS, IN ORDER TO        ADL48680
C EVALUATE PERSISTENCE OF THE TOXICANT AFTER THE EXTERNAL LOADINGS      ADL48690
C CEASE.  THE TIME-COURSE OF TOXICANT DISAPPEARANCE IS SUMMARIZED IN    ADL48700
C 'REPORT.DAT' AND IS WRITTEN TO A FILE ('KINE.DAT') USED FOR PLOTTING  ADL48710
C THE RESULTS.                                                          ADL48720
C-----------------------------------------------------------------------ADL48730
C                                                                       ADL48740
C     FEHLBERG FOURTH-FIFTH ORDER RUNGE-KUTTA METHOD                     ADL48750
C                                                                       ADL48760
C     WRITTEN BY H.A.WATTS AND L.F.SHAMPINE                             ADL48770
C                   SANDIA LABORATORIES                                 ADL48780
C                  ALBUQUERQUE,NEW MEXICO                               ADL48790
C                                                                       ADL48800
C       REWRITTEN FOR USE AT AERL OF EPA BY L.A. BURNS                  ADL48810
C                                                                       ADL48820
C    RKF45 IS PRIMARILY DESIGNED TO SOLVE NON-STIFF AND MILDLY STIFF    ADL48830
C    DIFFERENTIAL EQUATIONS WHEN DERIVATIVE EVALUATIONS ARE INEXPENSIVE.ADL48840
C    RKF45 SHOULD GENERALLY NOT BE USED WHEN THE USER IS DEMANDING      ADL48850
C    HIGH ACCURACY.                                                     ADL48860
C                                                                       ADL48870
C***********************************************************************ADL48880
C ABSTRACT                                                              ADL48890
C***********************************************************************ADL48900
C                                                                       ADL48910
C    SUBROUTINE  RKF45  INTEGRATES A SYSTEM OF NEQN FIRST ORDER         ADL48920
C    ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM                        ADL48930
C             DY(I)/DT = F(T,Y(1),Y(2),...,Y(NEQN))                     ADL48940
C               WHERE THE Y(I) ARE GIVEN AT T .                         ADL48950
C    TYPICALLY THE ROUTINE IS USED TO INTEGRATE FROM T TO TPRINT BUT IT ADL48960
C    CAN BE USED AS A ONE-STEP INTEGRATOR TO ADVANCE THE SOLUTION A     ADL48970
C    SINGLE STEP IN THE DIRECTION OF TPRINT.  ON RETURN                 ADL48980
C    THE CALL LIST IS SET FOR CONTINUING THE INTEGRATION. THE USER HAS  ADL48990
C    ONLY TO CALL RKF45 AGAIN AND PERHAPS DEFINE A NEW VALUE FOR TPRINT.ADL49000
C    ACTUALLY, RKF45 IS AN INTERFACING ROUTINE WHICH CALLS SUBROUTINE   ADL49010
C    RKFS FOR THE SOLUTION.  SUBSECTION FEHL OF RKFS COMPUTES AN        ADL49020
```

```
C    APPROXIMATE SOLUTION OVER ONE STEP.                                ADL49030
C      RKF45 HAS BEEN REWRITTEN (L.A.B.) TO SERVE AS A COMPLETE         ADL49040
C    INTERFACE TO RKFS, INCLUDING AN ANALYSIS OF THE RETURNED VALUES    ADL49050
C    OF IFLAG WITH APPROPRIATE REPAIRS.                                 ADL49060
C                                                                       ADL49070
C    RKF45  USES THE RUNGE-KUTTA-FEHLBERG (4,5)  METHOD DESCRIBED       ADL49080
C    IN THE REFERENCES                                                  ADL49090
C    E.FEHLBERG , LOW-ORDER CLASSICAL RUNGE-KUTTA FORMULAS WITH STEPSIZEADL49100
C                 CONTROL , NASA TR R-315.                              ADL49110
C                 ALSO IN  COMPUTING,6(1970),PP.61-71.                  ADL49120
C                                                                       ADL49130
C    L.F.SHAMPINE AND H.A.WATTS, PRACTICAL SOLUTION OF ORDINARY         ADL49140
C                 DIFFERENTIAL EQUATIONS BY RUNGE-KUTTA METHODS,        ADL49150
C                 SANDIA LABORATORIES REPORT SAND76-0585.               ADL49160
C                                                                       ADL49170
C    THE PERFORMANCE OF RKF45 IS ILLUSTRATED IN THE REFERENCE           ADL49180
C    L.F.SHAMPINE,H.A.WATTS,S.DAVENPORT, SOLVING NON-STIFF ORDINARY     ADL49190
C                 DIFFERENTIAL EQUATIONS-THE STATE OF THE ART ,         ADL49200
C                 SANDIA LABORATORIES REPORT SAND75-0182 .  ALSO IN     ADL49210
C                 SIAM REVIEW,18(1976),PP.376-411.                      ADL49220
C                                                                       ADL49230
C                                                                       ADL49240
C    THE PARAMETERS REPRESENT-                                          ADL49250
C      FCT: SUBROUTINE FCT(T,Y,YP) EVALUATES DERIVATIVES YP(I)=DY(I)/DT ADL49260
C      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED                     ADL49270
C      Y(*) -- SOLUTION VECTOR AT T                                     ADL49280
C      T -- INDEPENDENT VARIABLE                                        ADL49290
C      TPRINT -- OUTPUT POINT AT WHICH SOLUTION IS DESIRED              ADL49300
C      RELERR,ABSERR -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR LOCALADL49310
C            ERROR TEST. AT EACH STEP THE CODE REQUIRES THAT            ADL49320
C                 ABS(LOCAL ERROR) .LE. RELERR*ABS(Y) + ABSERR          ADL49330
C            FOR EACH COMPONENT OF THE LOCAL ERROR AND SOLUTION VECTORS ADL49340
C      IFLAG -- INDICATOR FOR STATUS OF INTEGRATION                     ADL49350
C      WORK(*) -- ARRAY TO HOLD INFORMATION INTERNAL TO RKF45 WHICH IS  ADL49360
C            NECESSARY FOR SUBSEQUENT CALLS. MUST BE DIMENSIONED        ADL49370
C            AT LEAST  3+6*NEQN                                         ADL49380
C      IWORK(*) -- INTEGER ARRAY USED TO HOLD INFORMATION INTERNAL TO   ADL49390
C            RKF45 WHICH IS NECESSARY FOR SUBSEQUENT CALLS. MUST BE     ADL49400
C            DIMENSIONED AT LEAST  5                                    ADL49410
C                                                                       ADL49420
C                                                                       ADL49430
C***********************************************************************ADL49440
C  FIRST CALL TO RKF45                                                  ADL49450
C***********************************************************************ADL49460
C                                                                       ADL49470
C    THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR THE ARRAYSADL49480
C    IN THE CALL LIST  -       Y(NEQN) , WORK(3+6*NEQN) , IWORK(5)  ,   ADL49490
C    DECLARE F IN AN EXTERNAL STATEMENT, SUPPLY SUBROUTINE F(T,Y,YP) ANDADL49500
C    INITIALIZE THE FOLLOWING PARAMETERS-                               ADL49510
C                                                                       ADL49520
C      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED.  (NEQN .GE. 1)     ADL49530
C      Y(*) -- VECTOR OF INITIAL CONDITIONS                             ADL49540
C      T -- STARTING POINT OF INTEGRATION , MUST BE A VARIABLE          ADL49550
C      TPRINT -- OUTPUT POINT AT WHICH SOLUTION IS DESIRED.             ADL49560
C        T=TPRINT IS ALLOWED ON THE FIRST CALL ONLY, IN WHICH CASE      ADL49570
C            RKF45 RETURNS WITH IFLAG=2 IF CONTINUATION IS POSSIBLE.    ADL49580
C      RELERR,ABSERR -- RELATIVE AND ABSOLUTE LOCAL ERROR TOLERANCES    ADL49590
```

```
C         WHICH MUST BE NON-NEGATIVE. RELERR MUST BE A VARIABLE WHILE   ADL49600
C         ABSERR MAY BE A CONSTANT. THE CODE SHOULD NORMALLY NOT BE     ADL49610
C         USED WITH RELATIVE ERROR CONTROL SMALLER THAN ABOUT 1.E-8 .   ADL49620
C         TO AVOID LIMITING PRECISION DIFFICULTIES THE CODE REQUIRES    ADL49630
C         RELERR TO BE LARGER THAN AN INTERNALLY COMPUTED RELATIVE      ADL49640
C         ERROR PARAMETER WHICH IS MACHINE DEPENDENT. IN PARTICULAR,    ADL49650
C         PURE ABSOLUTE ERROR IS NOT PERMITTED. IF A SMALLER THAN       ADL49660
C         ALLOWABLE VALUE OF RELERR IS ATTEMPTED, RKF45 INCREASES       ADL43670
C         RELERR APPROPRIATELY AND RETURNS CONTROL TO THE USER BEFORE   ADL49680
C         CONTINUING THE INTEGRATION.                                   ADL49690
C      IFLAG -- +1,-1  INDICATOR TO INITIALIZE THE CODE FOR EACH NEW    ADL49700
C         PROBLEM. NORMAL INPUT IS +1. THE USER SHOULD SET IFLAG=-1     ADL49710
C         ONLY WHEN ONE-STEP INTEGRATOR CONTROL IS ESSENTIAL. IN THIS   ADL49720
C         CASE, RKF45 ATTEMPTS TO ADVANCE THE SOLUTION A SINGLE STEP    ADL49730
C         IN THE DIRECTION OF TPRINT EACH TIME IT IS CALLED. SINCE THIS ADL49740
C         MODE OF OPERATION RESULTS IN EXTRA COMPUTING OVERHEAD, IT     ADL49750
C         SHOULD BE AVOIDED UNLESS NEEDED.                              ADL49760
C                                                                       ADL49770
C                                                                       ADL49780
C***********************************************************************ADL49790
C  OUTPUT FROM RKF45                                                    ADL49800
C***********************************************************************ADL49810
C                                                                       ADL49820
C      Y(*) -- SOLUTION AT T                                            ADL49830
C      T -- LAST POINT REACHED IN INTEGRATION.                          ADL49840
C      IFLAG=2 - INTEGRATION REACHED TPRINT. INDICATES SUCCESSFUL RETURNADL49850
C                   AND IS THE NORMAL MODE FOR CONTINUING INTEGRATION.  ADL49860
C            =-2 -- A SINGLE SUCCESSFUL STEP IN THE DIRECTION OF TPRINT ADL49870
C                   HAS BEEN TAKEN. NORMAL MODE FOR CONTINUING          ADL49880
C                   INTEGRATION ONE STEP AT A TIME.                     ADL49890
C            = 3 -- INTEGRATION WAS NOT COMPLETED BECAUSE RELATIVE ERRORADL49900
C                   TOLERANCE WAS TOO SMALL. RELERR HAS BEEN INCREASED  ADL49910
C                   APPROPRIATELY FOR CONTINUING.                       ADL49920
C            = 4 -- INTEGRATION WAS NOT COMPLETED BECAUSE MORE THAN     ADL49930
C                   3000 DERIVATIVE EVALUATIONS WERE NEEDED. THIS       ADL49940
C                   IS APPROXIMATELY 500 STEPS.                         ADL49950
C            = 5 -- INTEGRATION WAS NOT COMPLETED BECAUSE SOLUTION      ADL49960
C                   VANISHED MAKING A PURE RELATIVE ERROR TEST          ADL49970
C                   IMPOSSIBLE. MUST USE NON-ZERO ABSERR TO CONTINUE.   ADL49980
C                   USING THE ONE-STEP INTEGRATION MODE FOR ONE STEP    ADL49990
C                   IS A GOOD WAY TO PROCEED.                           ADL50000
C            = 6 -- INTEGRATION WAS NOT COMPLETED BECAUSE REQUESTED     ADL50010
C                   ACCURACY COULD NOT BE ACHIEVED USING SMALLEST       ADL50020
C                   ALLOWABLE STEPSIZE. USER MUST INCREASE THE ERROR    ADL50030
C                   TOLERANCE BEFORE CONTINUED INTEGRATION CAN BE       ADL50040
C                   ATTEMPTED.                                          ADL50050
C            = 7 -- IT IS LIKELY THAT RKF45 IS INEFFICIENT FOR SOLVING  ADL50060
C                   THIS PROBLEM. TOO MUCH OUTPUT IS RESTRICTING THE    ADL50070
C                   NATURAL STEPSIZE CHOICE. USE THE ONE-STEP INTEGRATORADL50080
C                   MODE.                                               ADL50090
C            = 8 -- INVALID INPUT PARAMETERS                            ADL50100
C                   THIS INDICATOR OCCURS IF ANY OF THE FOLLOWING IS    ADL50110
C                   SATISFIED -   NEQN .LE. 0                           ADL50120
C                               T=TPRINT  AND  IFLAG .NE. +1 OR -1      ADL50130
C                                 RELERR OR ABSERR .LT. 0.              ADL50140
C                                 IFLAG .EQ. 0  OR  .LT. -2  OR  .GT. 8 ADL50150
C      WORK(*),IWORK(*) -- INFORMATION WHICH IS USUALLY OF NO INTEREST   ADL50160
```

```
C                   TO THE USER BUT NECESSARY FOR SUBSEQUENT CALLS.    ADL50170
C                   WORK(1),...,WORK(NEQN) CONTAIN THE FIRST DERIVATIVESADL50180
C                   OF THE SOLUTION VECTOR Y AT T. WORK(NEQN+1) CONTAINSADL50190
C                   THE STEPSIZE H TO BE ATTEMPTED ON THE NEXT STEP.    ADL50200
C                   IWORK(1) CONTAINS THE DERIVATIVE EVALUATION COUNTER.ADL50210
C                                                                       ADL50220
C                                                                       ADL50230
C***********************************************************************ADL50240
C  SUBSEQUENT CALLS TO RKF45                                            ADL50250
C***********************************************************************ADL50260
C                                                                       ADL50270
C    SUBROUTINE RKF45 RETURNS WITH ALL INFORMATION NEEDED TO CONTINUE.  ADL50280
C     IF THE INTEGRATION REACHED TPRINT,THE USER NEED ONLY              ADL50290
C  DEFINE A NEW TPRINT AND CALL RKF45 AGAIN. IN THE ONE-STEP INTEGRATOR ADL50300
C    MODE (IFLAG=-2) THE USER MUST KEEP IN MIND THAT EACH STEP TAKEN IS ADL50310
C IN THE DIRECTION OF THE CURRENT TPRINT. UPON REACHING TPR (INDICATED ADL50320
C    BY CHANGING IFLAG TO 2),THE USER MUST THEN DEFINE A NEW TPRINT AND ADL50330
C    RESET IFLAG TO -2 TO CONTINUE IN THE ONE-STEP INTEGRATOR MODE.     ADL50340
C                                                                       ADL50350
C    IF THE INTEGRATION WAS NOT COMPLETED BUT THE USER STILL WANTS TO   ADL50360
C    CONTINUE (IFLAG=3,4 CASES), HE JUST CALLS RKF45 AGAIN. WITH IFLAG=3ADL50370
C    THE RELERR PARAMETER HAS BEEN ADJUSTED APPROPRIATELY FOR CONTINUINGADL50380
C    THE INTEGRATION. IN THE CASE OF IFLAG=4 THE FUNCTION COUNTER WILL  ADL50390
C    BE RESET TO 0 AND ANOTHER 3000 FUNCTION EVALUATIONS ARE ALLOWED.   ADL50400
C                                                                       ADL50410
C    HOWEVER,IN THE CASE IFLAG=5, THE USER MUST FIRST ALTER THE ERROR   ADL50420
C    CRITERION TO USE A POSITIVE VALUE OF ABSERR BEFORE INTEGRATION CAN ADL50430
C    PROCEED. IF HE DOES NOT,EXECUTION IS TERMINATED.                   ADL50440
C                                                                       ADL50450
C    ALSO,IN THE CASE IFLAG=6, IT IS NECESSARY FOR THE USER TO RESET    ADL50460
C    IFLAG TO 2 (OR -2 WHEN THE ONE-STEP INTEGRATION MODE IS BEING USED)ADL50470
C    AS WELL AS INCREASING EITHER ABSERR,RELERR OR BOTH BEFORE THE      ADL50480
C    INTEGRATION CAN BE CONTINUED. IF THIS IS NOT DONE, EXECUTION WILL  ADL50490
C    BE TERMINATED. THE OCCURRENCE OF IFLAG=6 INDICATES A TROUBLE SPOT  ADL50500
C    (SOLUTION IS CHANGING RAPIDLY,SINGULARITY MAY BE PRESENT) AND IT   ADL50510
C    OFTEN IS INADVISABLE TO CONTINUE.                                  ADL50520
C                                                                       ADL50530
C    IF IFLAG=7 IS ENCOUNTERED, THE USER SHOULD USE THE ONE-STEP        ADL50540
C    INTEGRATION MODE WITH THE STEPSIZE DETERMINED BY THE CODE OR       ADL50550
C    CONSIDER SWITCHING TO THE ADAMS CODES DE/STEP,INTRP. IF THE USER   ADL50560
C    INSISTS UPON CONTINUING THE INTEGRATION WITH RKF45, HE MUST RESET  ADL50570
C    IFLAG TO 2 BEFORE CALLING RKF45 AGAIN. OTHERWISE,EXECUTION WILL BE ADL50580
C    TERMINATED.                                                        ADL50590
C                                                                       ADL50600
C    IF IFLAG=8 IS OBTAINED, INTEGRATION CANNOT BE CONTINUED UNLESS     ADL50610
C    THE INVALID INPUT PARAMETERS ARE CORRECTED.                        ADL50620
C                                                                       ADL50630
C    IT SHOULD BE NOTED THAT THE ARRAYS WORK,IWORK CONTAIN INFORMATION  ADL50640
C    REQUIRED FOR SUBSEQUENT INTEGRATION. ACCORDINGLY, WORK AND IWORK   ADL50650
C    SHOULD NOT BE ALTERED.                                             ADL50660
C                                                                       ADL50670
C***********************************************************************ADL50680
C                                                                       ADL50690
C***********************************************************************ADL50700
C FILE WITH GLOBAL COMMONS DESCRIBING THE TOXICANT AND ENVIRONMENT:     ADL50710
      REAL MWTG,SOLG,ESOLG,PKBG,EPKBG,PKAG,EPKAG,KPSG,KPBG,KOCG,KOWG    ADL50720
     1,KCECG,KAECG,KVOG,HENRYG,VAPRG,EVPRG,QUANTG,KDPG,RFLATG,ABSG,KAHG ADL50730
     1,EAHG,KNHG,ENHG,KBHG,EBHG,KOXG,EOXG,KBACWG,QTBAWG,KBACSG,QTBASG   ADL50740
     1,STRLDG,NPSLDG,PCPLDG,DRFLDG,IFLLDG,ADVPRG,XSTURG,CHARLG,DSPG     ADL50750
     1,SDCHRG,FROCG,CECG,AECG,PCTWAG,TCELG,PHG,POHG,OXRADG,BIOMSG,PLRAG ADL50760
     1,BIOTMG,BACTOG,ACBACG,KO2G,CMPETG,DOCG,CHLG,WLAMG,CLOUDG,DFACG    ADL50770
     1,VOLG,AREAG,DEPTHG,RAING,EVAPG,LATG,WINDG,STFLOG,STSEDG,NPSFLG    ADL50780
     1,NPSEDG,INTFLG                                                    ADL50790
      INTEGER    SPFLG,TYPEE,KOUNT,JFRADG,ITOADG,JTURBG,ITURBG,BATCH    ADL50800
      INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT                          ADL50810
      LOGICAL*1 CHEMNA,SYSTYP                                           ADL50820
      COMMON/NAME1G/CHEMNA(60)                                          ADL50830
      COMMON/CONTRG/SPFLG(5)                                            ADL50840
      COMMON/PCHEMG/MWTG,SOLG(5),ESOLG(5)                               ADL50850
      COMMON/IONG/PKBG(2),EPKBG(2),PKAG(2),EPKAG(2)                     ADL50860
      COMMON/PARTG/KPSG(5),KPBG(5),KOCG,KOWG,KCECG(2),KAECG(2)          ADL50870
      COMMON/VOLATG/KVOG,HENRYG,VAPRG,EVPRG                             ADL50880
      COMMON/DPHOTG/QUANTG(3,5),KDPG(5),RFLATG(5),ABSG(39,5)            ADL50890
      COMMON/HYDROG/KAHG(3,5),EAHG(3,5),KNHG(3,5),ENHG(3,5)             ADL50900
     1,KBHG(3,5),EBHG(3,5)                                              ADL50910
      COMMON/OXIDG/KOXG(3,5),EOXG(3,5)                                  ADL50920
      COMMON/BACTG/KBACWG(3,5),QTBAWG(3,5),KBACSG(3,5),QTBASG(3,5)      ADL50930
      COMMON/LOADSG/STRLDG(10),NPSLDG(10),PCPLDG(10),DRFLDG(10)         ADL50940
     1,IFLLDG(10)                                                       ADL50950
      COMMON/NAME2G/SYSTYP(60),TYPEE(10)                                ADL50960
      COMMON/SETUPG/KOUNT,JFRADG(30),ITOADG(30),ADVPRG(30),JTURBG(30)   ADL50970
     1,ITURBG(30),XSTURG(30),CHARLG(30),DSPG(30),BATCH                  ADL50980
      COMMON/SEDMG/SDCHRG(10),FROCG(10),CECG(10),AECG(10),PCTWAG(10)    ADL50990
      COMMON/QUALG/TCELG(10),PHG(10),POHG(10),OXRADG(10),BIOMSG(10)     ADL51000
     1,PLRAG(10),BIOTMG(10),BACTOG(10),ACBACG(10),KO2G(10)              ADL51010
      COMMON/PHOG/CMPETG(10),DOCG(10),CHLG(10),WLAMG(39),CLOUDG         ADL51020
     1,DFACG(10)                                                        ADL51030
      COMMON/GEOMT/VOLG(10),AREAG(10),DEPTHG(10)                        ADL51040
      COMMON/CLIMG/RAING,EVAPG(10),LATG,WINDG(10)                       ADL51050
      COMMON/FLOWG/STFLOG(10),STSEDG(10),NPSFLG(10),NPSEDG(10)          ADL51060
     1,INTFLG(10)                                                       ADL51070
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT        ADL51080
C FILE WITH LOCAL COMMONS DESCRIBING THE TOXICANT:                      ADL51090
C---------------------------------------------------------------------- ADL51100
C FILE 'CHEML.COM'                                                      ADL51110
        COMMON/PART1L/KOCL,KPSL(5,10),ALPHA(18,10),YSATL(5,10)          ADL51120
        COMMON/CHEM1L/BIOLKL(10),HYDRKL(10),OXIDKL(10),PHOTKL(10),      ADL51130
     2  VOLKL(10),EXPOKL(10),TOTKL(10),INTINL(10,10)                    ADL51140
        COMMON/MISCL/SYSLDL,CONLDL(10),TOTLDL(10)                       ADL51150
        COMMON/RESULT/Z(18),QWSAV,QSSAV,QTSAV                           ADL51160
        COMMON/RESLT/DOMAX(10),MAXPT(10),DOMIN(10),MINPT(10),           ADL51170
     2  CHEMPC,BIOPCT,VOLPCT,EXPPCT                                     ADL51180
        COMMON/CHEM2L/KB1L(10),KB2L(10),KA1L(10),KA2L(10)               ADL51190
        REAL KOCL,KPSL,INTINL,KB1L,KB2L,KA1L,KA2L                       ADL51200
C---------------------------------------------------------------------- ADL51210
C FILE WITH LOCAL COMMONS DESCRIBING THE ENVIRONMENT:                   ADL51220
C---------------------------------------------------------------------- ADL51230
C FILE 'ENVIRL.COM'                                                     ADL51240
        COMMON/DPHOTL/WATETA(39),PIGETA(39),DOCETA(39)                  ADL51250
        COMMON/FLOWSL/STRMFL(10),STSCOL(10),RAINFL(10),INTFL(10),       ADL51260
     2  NPSFL(10),NPSCOL(10),EVAPL(10),TOTEVL                           ADL51270
        COMMON/RKFL/T,TFINAL,TINCR,TPRINT,ABSERR,RELERR,IFLAG,          ADL51280
     2  KDTIME,TFACTR,IPER,PERIOD                                       ADL51290
        COMMON/SETUPL/L,E,H,B,KOUNTW,KOUNTS,INDEXW(10),INDEXS(10)       ADL51300
     1,JSAV1,JSAV2,ICALL,NPX,NCON                                       ADL51310
        COMMON/MASSL/WATVOL(10),SEDMSL(10),SEDCOL(10),BIOTOL(10),       ADL51320
     1     PSEDCO(10)                                                   ADL51330
        COMMON/FLOWL/WATOUL(10),SEDOUL(10),WATFL(10,10),                ADL51340
     2 SEDFL(10,10),WATINL(10)                                          ADL51350
        COMMON/ENPARL/KO2L(10),LIGHTL(10),BOTLIT,BOTLAM(39),            ADL51360
     1  WAVEL(39),TOTETA(39),WLAML(39)                                  ADL51370
        REAL INTFL,NPSFL,NPSCOL                                         ADL51380
        REAL KO2L,KOUNTW,KOUNTS,LIGHTL                                  ADL51390
        LOGICAL PRINT                                                   ADL51400
        INTEGER L,E,H,B                                                 ADL51410
C-------------------------------------------------------------------    ADL51420
C***********************************************************************ADL51430
C DIMENSION WORK(63) FOR 10 COMPARTMENTS (NEQN=10 AT MOST)              ADL51440
      DIMENSION Y(1),WORK(63),IWORK(5)                                  ADL51450
C                                                                       ADL51460
      EXTERNAL FCT,OUTP                                                 ADL51470
C                                                                       ADL51480
C CONTROL TIME SEQUENCE:                                                ADL51490
        LOGICAL LFLAG                                                   ADL51500
C                                                                       ADL51510
C CONTROL NORMAL VS. ONE-STEP METHOD                                    ADL51520
        LOGICAL MEFLAG                                                  ADL51530
C       LFLAG IS CONTROL FLAG FOR INTEGRATION TIMES, MEFLAG IS          ADL51540
C       CONTROL FLAG FOR MODE OF OPERATION (NORMAL OR ONE-STEP).        ADL51550
C                                                                       ADL51560
C     COMPUTE INDICES FOR THE SPLITTING OF THE WORK ARRAY               ADL51570
C                                                                       ADL51580
C NEQN = NUMBER OF ACTIVE COMPARTMENTS                                  ADL51590
      NEQN=KOUNT                                                        ADL51600
      K1M=NEQN+1                                                        ADL51610
      K1=K1M+1                                                          ADL51620
      K2=K1+NEQN                                                        ADL51630
      K3=K2+NEQN                                                        ADL51640
      K4=K3+NEQN                                                        ADL51650
      K5=K4+NEQN                                                        ADL51660
      K6=K5+NEQN                                                        ADL51670
C***********************************************************************ADL51680
C     THIS INTERFACING ROUTINE MERELY RELIEVES THE USER OF A LONG       ADL51690
C     CALLING LIST VIA THE SPLITTING APART OF TWO WORKING STORAGE       ADL51700
C     ARRAYS. IF THIS IS NOT COMPATIBLE WITH THE USERS COMPILER,        ADL51710
C     HE MUST USE RKFS DIRECTLY.                                        ADL51720
C***********************************************************************ADL51730
C                                                                       ADL51740
C                                                                       ADL51750
C                                                                       ADL51760
C       INITIALIZATION OF CONTROL VARIABLES:                            ADL51770
C NUMBER OF BLOCKS OF OUTPUT PRINTED: ASSUMED IFLAG ON INPUT IS 1 OR -1 ADL51780
        NPRINT=-1                                                       ADL51790
C NUMBER OF BLOCKS OF MAXNFE FUNCTION EVALUATIONS:                      ADL51800
        NBLFE=0                                                         ADL51810
C                                                                       ADL51820
C TESTS FOR INABILITY TO PROCEED:                                       ADL51830
         TEST1=0.                                                       ADL51840
         TEST2=-1.                                                      ADL51850
C SINGULARITY TESTS:                                                    ADL51860
        TEST3=0.                                                        ADL51870
        TEST4=0.
C
C TIMER CONTROL:
        LFLAG=.TRUE.
C TEST FOR NEAR HIT ON TFINAL:
        EPINCR=0.001*TINCR
C NORMAL MODE (I.E. NOT ONE-STEP)
        MEFLAG=.FALSE.
C FOR ONE-STEP MODE:
         IF (IFLAG.EQ.-1) MEFLAG=.TRUE.
100   CALL RKFS(FCT,NEQN,Y,T,TPRINT,RELERR,ABSERR,IFLAG,WORK(1),
     1 WORK(K1M),WORK(K1),WORK(K2),WORK(K3),WORK(K4),WORK(K5),WORK(K6),
     2   WORK(K6+1),IWORK(1),IWORK(2),IWORK(3),IWORK(4),IWORK(5))
C
C
C
C IF IN ONE-STEP MODE:
        IF (MEFLAG) GO TO 110
C
C IF NOT IN ONE-STEP MODE:
        GO TO 120
C
C       DIRECTIONS FOR CONTINUING ONE-STEP INTEGRATION:
C
C CONTINUE ON -
110     IF(IFLAG.EQ.-2 .AND. T.LT.TPRINT) GO TO 100
        IF(IFLAG.EQ.2 .AND. T.EQ.TPRINT) SAVTPR=TPRINT
        IF(IFLAG.EQ.2 .AND. T.EQ.TPRINT) CALL OUTP(Y,WORK)
C
C INVESTIGATION OF IFLAG
        IF(IFLAG.NE.2 .AND. IFLAG.NE.-2) GO TO 120
C PROCEED WITH ONE-STEP MODE TO NEXT PRINT TIME
        IFLAG= -2
C
C SET UP NEXT TIME AND CHECK TFINAL:
        GO TO 170
C
C
C TOTAL DISASTER?
120     IF(IFLAG.EQ.0 .OR. IFLAG.EQ.-1) GO TO 130
C
C RETURN ON INVOCATION OF ONE-STEP BY A PROBLEM:
        IF(IFLAG.EQ.-2) GO TO 150
C
C
C ANALYZE SIGNAL:
        GO TO (130,160,220,240,290,310,370,390,410),IFLAG
C
C RKFS CRUMBLED
130     WRITE(TTYOUT,140)
140     FORMAT(1X,'INTEGRATOR IS NOT EXECUTING PROPERLY --'/1X,
     1  'SIMULATION ABORTED.  -- SORRY.')
        IFLAG=9
        GO TO 410
C
C
C
```

```
C       NORMAL RETURN: IFLAG=2 OR IFLAG=-2                              ADL52450
C                                                                       ADL52460
C       IF THE OUTPUT SEQUENCE DOES NOT INCLUDE THE ENDPOINT            ADL52470
C         (I.E. INCREMENTING T BY TINCR MISSES TFINAL)                  ADL52480
C       THE PROGRAM AUTOMATICALLY EXCLUDES OUTPUT BEYOND THE            ADL52490
C       ENDPOINT (T>TFINAL) AND GIVES OUTPUT FOR TFINAL                 ADL52500
C                                                                       ADL52510
C                                                                       ADL52520
C IFLAG RESET AFTER INVOCATION OF SINGLE-STEP MODE BY A PROBLEM         ADL52530
150     IFLAG=2                                                         ADL52540
        IF(T.LT.TPRINT) GO TO 100                                       ADL52550
C                                                                       ADL52560
C NORMAL CONTINUATION, TPRINT MAY BE CHANGED IN OUTP:                   ADL52570
160     SAVTPR=TPRINT                                                   ADL52580
        NPRINT=NPRINT+1                                                 ADL52590
        CALL OUTP(Y,WORK)                                               ADL52600
C                                                                       ADL52610
C IF TPRINT WAS INCREMENTED FROM OUTP:                                  ADL52620
170     IF(SAVTPR.LT.TPRINT) GO TO 190                                  ADL52630
C                                                                       ADL52640
C FOR INTEGRATION OVER A SINGLE INTERVAL WITHOUT                        ADL52650
C ANY INTERMEDIATE PRINT POINTS.                                        ADL52660
        IF(TPRINT.EQ.PERIOD) GO TO 410                                  ADL52670
        IF((PERIOD-TPRINT).LT.TINCR) TPRINT=PERIOD-TINCR                ADL52680
180     TPRINT=TPRINT+TINCR                                             ADL52690
C                                                                       ADL52700
C NEAR ENDPOINT?                                                        ADL52710
190     IF(ABS(PERIOD-TPRINT).GT.EPINCR) GO TO 210                      ADL52720
C                                                                       ADL52730
C WITHIN 0.1% IF ARRIVED HERE VIA PRECEDING LINE                        ADL52740
200     TPRINT=PERIOD                                                   ADL52750
C                                                                       ADL52760
C DISABLE LOOP:                                                         ADL52770
        LFLAG=.FALSE.                                                   ADL52780
C                                                                       ADL52790
C GO BACK TO GET Y AT TFINAL EXACTLY:                                   ADL52800
        GO TO 100                                                       ADL52810
C                                                                       ADL52820
C STANDARD CONTINUATION:                                                ADL52830
210     IF (TPRINT.LE.PERIOD) GO TO 100                                 ADL52840
C                                                                       ADL52850
C TINCR OVERSHOT THE NOMINAL ENDPOINT:                                  ADL52860
        IF (LFLAG) GO TO 200                                            ADL52870
C                                                                       ADL52880
C INTEGRATION COMPLETE TO TFINAL:                                       ADL52890
        GO TO 410                                                       ADL52900
C                                                                       ADL52910
C                                                                       ADL52920
220     CONTINUE                                                        ADL52930
C RELERR HAS BEEN RESET TO A VALUE APPROPRIATE TO THE MACHINE.          ADL52940
        GO TO 100                                                       ADL52950
C                                                                       ADL52960
C                                                                       ADL52970
C       IFLAG=4                                                         ADL52980
C                                                                       ADL52990
240     TEST1=T                                                         ADL53000
C INCREMENT COUNTER FOR NUMBER OF BLOCKS EXECUTED:                      ADL53010
```

```
      NBLFE=NBLFE+1                                                     ADL53020
C                                                                       ADL53030
C NO PROGRESS POSSIBLE?                                                  ADL53040
      IF(TEST1.EQ.TEST2) GO TO 270                                      ADL53050
      TEST2=TEST1                                                       ADL53060
      IF (NBLFE.LT.5) GO TO 250                                         ADL53070
C TIMEOUT: TERMINATE KINETIC SIMULATION -                                ADL53080
      TITEST=TINCR*(FLOAT(NPRINT))                                      ADL53090
      IF (TITEST.GT.0.) GO TO 245                                       ADL53100
C SIMULATION STILL ON INITIAL CONDITION:                                ADL53110
      WRITE (TTYOUT,241)                                                ADL53120
241   FORMAT(/1X,'THE KINETIC EQUATIONS ARE TOO INTRACTABLE FOR THIS'   ADL53130
     1,' PROGRAM.')                                                     ADL53140
      GO TO 405                                                         ADL53150
C SOME KINETIC DATA AVAILABLE:                                          ADL53160
245   WRITE(TTYOUT,246)                                                 ADL53170
246   FORMAT(/1X,'THE KINETIC EQUATIONS ARE INCONVENIENTLY STIFF;'      ADL53180
     1/1X,'THE DECAY SIMULATION HAS BEEN ABBREVIATED.')                 ADL53190
      IF (TITEST.LT.PERIOD) PERIOD=TITEST                               ADL53200
      GO TO 410                                                         ADL53210
250   WRITE(TTYOUT,260)                                                 ADL53220
260   FORMAT(/1X,'PLEASE BE PATIENT.  THIS IS TAKING LONGER',           ADL53230
     1 1X,'THAN I EXPECTED.')                                           ADL53240
      GO TO 100                                                         ADL53250
C                                                                       ADL53260
270   WRITE(TTYOUT,280)                                                 ADL53270
280   FORMAT(1X,'THE INTEGRATOR IS NOT MAKING ANY PROGRESS.'//1X,       ADL53280
     1 'THE SIMULATION HAS BEEN ABORTED.  -- SORRY.')                   ADL53290
      IFLAG=9                                                           ADL53300
      GO TO 410                                                         ADL53310
C                                                                       ADL53320
C                                                                       ADL53330
C     IFLAG=5                                                           ADL53340
C                                                                       ADL53350
290   CONTINUE                                                          ADL53360
C PURE RELATIVE ERROR TEST NO GOOD - SOLUTION VANISHED.                 ADL53370
      ABSERR=1.E-06                                                     ADL53380
      IFLAG=-2                                                          ADL53390
      GO TO 100                                                         ADL53400
C                                                                       ADL53410
C                                                                       ADL53420
C     IFLAG=6                                                           ADL53430
C                                                                       ADL53440
310   CONTINUE                                                          ADL53450
C ACCURACY REQUESTED NOT ACHIEVABLE WITH SMALLEST ALLOWABLE             ADL53460
C STEPSIZE - POSSIBLE SINGULARITY.                                      ADL53470
      TEST3=T                                                           ADL53480
C                                                                       ADL53490
C SINGULARITY?                                                          ADL53500
      IF(TEST3.EQ.TEST4) GO TO 350                                      ADL53510
      TEST4=TEST3                                                       ADL53520
C INCREASE ERROR TOLERANCES AND TRY AGAIN:                              ADL53530
      ABSERR=2.*ABSERR+1.E-08                                           ADL53540
      RELERR=2.*RELERR+1.E-08                                           ADL53550
C                                                                       ADL53560
C IF IN ONE-STEP MODE:                                                  ADL53570
      IF(MEFLAG) GO TO 340                                              ADL53580
```

```
           IFLAG=2                                                      ADL53590
           GO TO 100                                                    ADL53600
 340       IFLAG=-2                                                     ADL53610
           GO TO 100                                                    ADL53620
 350       WRITE(TTYOUT,360)                                            ADL53630
 360       FORMAT(1X,'THE INTEGRATOR HAS DETECTED A MATHEMATICAL',      ADL53640
     1 1X,'SINGULARITY.'/1X,'SIMULATION DISCONTINUED.')                 ADL53650
           IFLAG=9                                                      ADL53660
           GO TO 410                                                    ADL53670
C                                                                       ADL53680
C                                                                       ADL53690
C          IFLAG=7                                                      ADL53700
C                                                                       ADL53710
 370       CONTINUE                                                     ADL53720
C TOO MUCH OUTPUT IS RESTRICTING THE NATURAL STEP-SIZE.                 ADL53730
C SWITCH TO ONE-STEP MODE.                                              ADL53740
           MEFLAG=.TRUE.                                                ADL53750
           IFLAG=-2                                                     ADL53760
           GO TO 100                                                    ADL53770
C                                                                       ADL53780
C                                                                       ADL53790
C          IFLAG=8                                                      ADL53800
C                                                                       ADL53810
 390       WRITE(TTYOUT,400)                                            ADL53820
 400       FORMAT(1X,'THE INTEGRATOR RECEIVED AN INVALID INPUT AND',    ADL53830
     1 1X,'WAS UNABLE TO PROCEED.'/1X,'SIMULATION WAS DISCONTINUED.'/   ADL53840
     2 1X,'I SUGGEST THAT YOU EXIT AND RESTART EXAMS.')                 ADL53850
 405       IFLAG=9                                                      ADL53860
 410    RETURN                                                          ADL53870
        END                                                             ADL53880
C FILE 'RKFS.FTN'                                                       ADL53890
        SUBROUTINE RKFS(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,YP,HH,F1,F2, ADL53900
     1              F3,F4,F5,SAVRE,SAVAE,NFE,KOP,INIT,JFLAG,KFLAG)      ADL53910
C                                                                       ADL53920
C       FEHLBERG FOURTH-FIFTH ORDER RUNGE-KUTTA METHOD                  ADL53930
C                                                                       ADL53940
C***********************************************************************ADL53950
C                                                                       ADL53960
C       RKFS INTEGRATES A SYSTEM OF FIRST ORDER ORDINARY DIFFERENTIAL   ADL53970
C       EQUATIONS AS DESCRIBED IN THE COMMENTS FOR RKF45 .              ADL53980
C       THE ARRAYS YP,F1,F2,F3,F4,AND F5 (OF DIMENSION AT LEAST NEQN) AND ADL53990
C       THE VARIABLES H,SAVRE,SAVAE,NFE,KOP,INIT,JFLAG,AND KFLAG ARE USED ADL54000
C       INTERNALLY BY THE CODE AND APPEAR IN THE CALL LIST TO ELIMINATE ADL54010
C       LOCAL RETENTION OF VARIABLES BETWEEN CALLS. ACCORDINGLY, THEY   ADL54020
C       SHOULD NOT BE ALTERED. ITEMS OF POSSIBLE INTEREST ARE           ADL54030
C           YP - DERIVATIVE OF SOLUTION VECTOR AT T                     ADL54040
C           H  - AN APPROPRIATE STEPSIZE TO BE USED FOR THE NEXT STEP   ADL54050
C           NFE- COUNTER ON THE NUMBER OF DERIVATIVE FUNCTION EVALUATIONS ADL54060
C                                                                       ADL54070
C***********************************************************************ADL54080
C                                                                       ADL54090
        LOGICAL HFAILD,OUTPUT                                           ADL54100
C                                                                       ADL54110
        DIMENSION Y(NEQN),YP(NEQN),F1(NEQN),F2(NEQN),F3(NEQN),F4(NEQN), ADL54120
     1            F5(NEQN)                                              ADL54130
C                                                                       ADL54140
        EXTERNAL F                                                      ADL54150
```

```
      COMMON/UNITS/RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT,PRINT,NPRINT
        INTEGER RPTOUT,SSOUT,KINOUT,TTYIN,TTYOUT
        LOGICAL PRINT
C
C ------------------------------------------------------------------------
C THE COMPUTER UNIT ROUNDOFF ERROR (U), AND THE MINIMUM VALUE FOR THE
C RELATIVE ERROR TOLERANCE (REMIN) ARE MACHINE-DEPENDENT CONSTANTS.
C   THE COMPUTER UNIT ROUNDOFF ERROR 'U' IS THE SMALLEST POSITIVE VALUE
C REPRESENTABLE IN THE MACHINE SUCH THAT (1.+U).GT.1.
C   REMIN IS A TOLERANCE THRESHOLD WHICH IS ALSO DETERMINED BY THE
C INTEGRATION METHOD.  IN PARTICULAR, A FIFTH-ORDER METHOD WILL
C GENERALLY NOT BE CAPABLE OF DELIVERING ACCURACIES NEAR LIMITING
C PRECISION ON COMPUTERS WITH LONG WORDLENGTHS.  REMIN SHOULD NOT BE
C SMALLER THAN 4.*U; AND SHOULD BE SPECIFIED AS THE LARGER OF 4.*U AND
C 1.E-12.  VALUES FOR SOME REPRESENTATIVE MACHINES ARE:
C
C      MACHINE              U              REMIN
C  ---------------     ------------     -----------
C  IBM 360/370 SP        9.54E-7          3.82E-6
C  IBM 360/370 DP        2.22D-16         1.D-12
C  PDP-11      SP        5.96E-8          2.39E-7
C  PDP-11      DP        1.39E-17         1.D-12
C  PDP-10                7.45E-9          2.99E-8
C  CDC 6000 SERIES       7.11E-15         1.E-12
C  UNIVAC 1108           1.49E-8          5.97E-8
C
      DATA U/9.54E-7/,REMIN/3.82E-6/
C-------------------------------------------------------------------------
C
C     THE EXPENSE IS CONTROLLED BY RESTRICTING THE NUMBER
C     OF FUNCTION EVALUATIONS TO BE APPROXIMATELY MAXNFE.
C     AS SET,THIS CORRESPONDS TO ABOUT 500 STEPS.
C
      DATA MAXNFE/3000/
C
C*************************************************************************
C
C
C     CHECK INPUT PARAMETERS
C
C
      IF (NEQN .LT. 1) GO TO 110
      IF ((RELERR .LT. 0.)  .OR.  (ABSERR .LT. 0.)) GO TO 130
      MFLAG=IABS(IFLAG)
      IF ((MFLAG .GE. 1)  .AND.  (MFLAG .LE. 8)) GO TO 150
C
C     INVALID INPUT FLAG
100   IFLAG=8
      GO TO 440
C
110     WRITE(TTYOUT,120)
120     FORMAT(1X,'NO EQUATIONS SPECIFIED!')
        IFLAG=9
        GO TO 440
130     WRITE(TTYOUT,140)
140     FORMAT(1X,'NEGATIVE ERROR CRITERIA CANNOT WORK!')
        IFLAG=9
        GO TO 440                                                       ADL54730
C     IS THIS THE FIRST CALL?                                           ADL54740
150   IF (MFLAG .EQ. 1) GO TO 200                                       ADL54750
C                                                                       ADL54760
C     CHECK CONTINUATION POSSIBILITIES                                  ADL54770
C                                                                       ADL54780
      IF ((T .EQ. TOUT) .AND. (KFLAG .NE. 3)) GO TO 100                 ADL54790
      IF (MFLAG .NE. 2) GO TO 160                                       ADL54800
C                                                                       ADL54810
C     IFLAG = +2 OR -2                                                  ADL54820
      IF (KFLAG .EQ. 3) GO TO 190                                       ADL54830
      IF (INIT .EQ. 0) GO TO 190                                        ADL54840
      IF (KFLAG .EQ. 4) GO TO 180                                       ADL54850
      IF ((KFLAG .EQ. 5)  .AND.  (ABSERR .EQ. 0.)) GO TO 170            ADL54860
      IF ((KFLAG .EQ. 6)  .AND.  (RELERR .LE. SAVRE)  .AND.             ADL54870
     1    (ABSERR .LE. SAVAE)) GO TO 170                                ADL54880
      GO TO 200                                                         ADL54890
C                                                                       ADL54900
C     IFLAG = 3,4,5,6,7 OR 8                                            ADL54910
160   IF (IFLAG .EQ. 3) GO TO 190                                       ADL54920
      IF (IFLAG .EQ. 4) GO TO 180                                       ADL54930
      IF ((IFLAG .EQ. 5) .AND. (ABSERR .GT. 0.)) GO TO 190              ADL54940
C                                                                       ADL54950
C     INTEGRATION CANNOT BE CONTINUED SINCE USER DID NOT RESPOND TO     ADL54960
C     THE INSTRUCTIONS PERTAINING TO IFLAG=5,6,7 OR 8                   ADL54970
170    IFLAG=9                                                          ADL54980
        GO TO 440                                                       ADL54990
C                                                                       ADL55000
C***********************************************************************ADL55010
C                                                                       ADL55020
C     RESET FUNCTION EVALUATION COUNTER                                 ADL55030
180   NFE=0                                                             ADL55040
      IF (MFLAG .EQ. 2) GO TO 200                                       ADL55050
C                                                                       ADL55060
C     RESET FLAG VALUE FROM PREVIOUS CALL                               ADL55070
190   IFLAG=JFLAG                                                       ADL55080
      IF (KFLAG .EQ. 3) MFLAG=IABS(IFLAG)                               ADL55090
C                                                                       ADL55100
C     SAVE INPUT IFLAG AND SET CONTINUATION FLAG VALUE FOR SUBSEQUENT   ADL55110
C     INPUT CHECKING                                                    ADL55120
200   JFLAG=IFLAG                                                       ADL55130
      KFLAG=0                                                           ADL55140
C                                                                       ADL55150
C     SAVE RELERR AND ABSERR FOR CHECKING INPUT ON SUBSEQUENT CALLS     ADL55160
      SAVRE=RELERR                                                      ADL55170
      SAVAE=ABSERR                                                      ADL55180
C                                                                       ADL55190
C     RESTRICT RELATIVE ERROR TOLERANCE TO BE AT LEAST AS LARGE AS      ADL55200
C     2U+REMIN TO AVOID LIMITING PRECISION DIFFICULTIES ARISING FROM    ADL55210
C     IMPOSSIBLE ACCURACY REQUESTS                                      ADL55220
C                                                                       ADL55230
      RER=2.*U+REMIN                                                    ADL55240
      IF (RELERR .GE. RER) GO TO 210                                    ADL55250
C                                                                       ADL55260
C     RELATIVE ERROR TOLERANCE TOO SMALL                                ADL55270
      RELERR=RER                                                        ADL55280
      IFLAG=3                                                           ADL55290
      KFLAG=3                                                           ADL55300
      GO TO 440                                                         ADL55310
C                                                                       ADL55320
210   U26=26.*U                                                         ADL55330
C                                                                       ADL55340
      DT=TOUT-T                                                         ADL55350
C                                                                       ADL55360
      IF (MFLAG .EQ. 1) GO TO 220                                       ADL55370
      IF (INIT .EQ. 0) GO TO 230                                        ADL55380
      GO TO 250                                                         ADL55390
C                                                                       ADL55400
C                                                                       ADL55410
C***********************************************************************ADL55420
C                                                                       ADL55430
C     INITIALIZATION --                                                 ADL55440
C                                                                       ADL55450
C                       SET INITIALIZATION COMPLETION INDICATOR,INIT    ADL55460
C                       SET INDICATOR FOR TOO MANY OUTPUT POINTS,KOP    ADL55470
C                       EVALUATE INITIAL DERIVATIVES                    ADL55480
C                       SET COUNTER FOR FUNCTION EVALUATIONS,NFE        ADL55490
C                       ESTIMATE STARTING STEPSIZE                      ADL55500
220   INIT=0                                                            ADL55510
      KOP=0                                                             ADL55520
C                                                                       ADL55530
      A=T                                                               ADL55540
      CALL F(A,Y,YP)                                                    ADL55550
      NFE=1                                                             ADL55560
      IF (T .NE. TOUT) GO TO 230                                        ADL55570
      IFLAG=2                                                           ADL55580
      GO TO 440                                                         ADL55590
C                                                                       ADL55600
C                                                                       ADL55610
230   INIT=1                                                            ADL55620
      HH=ABS(DT)                                                        ADL55630
      TOLN=0.                                                           ADL55640
      DO 240 K=1,NEQN                                                   ADL55650
        TOL=RELERR*ABS(Y(K))+ABSERR                                     ADL55660
        IF (TOL .LE. 0.) GO TO 240                                      ADL55670
        TOLN=TOL                                                        ADL55680
        YPK=ABS(YP(K))                                                  ADL55690
        IF (YPK*HH**5 .GT. TOL) HH=(TOL/YPK)**0.2                       ADL55700
240   CONTINUE                                                          ADL55710
      IF (TOLN .LE. 0.) HH=0.                                           ADL55720
      HH=AMAX1(HH,U26*AMAX1(ABS(T),ABS(DT)))                            ADL55730
      JFLAG=ISIGN(2,IFLAG)                                              ADL55740
C                                                                       ADL55750
C                                                                       ADL55760
C***********************************************************************ADL55770
C                                                                       ADL55780
C     SET STEPSIZE FOR INTEGRATION IN THE DIRECTION FROM T TO TOUT      ADL55790
C                                                                       ADL55800
250   HH=SIGN(HH,DT)                                                    ADL55810
C                                                                       ADL55820
C     TEST TO SEE IF RKF45 IS BEING SEVERELY IMPACTED BY TOO MANY       ADL55830
C     OUTPUT POINTS                                                     ADL55840
C                                                                       ADL55850
      IF (ABS(HH) .GE. 2.*ABS(DT)) KOP=KOP+1                            ADL55860
      IF (KOP .NE. 100) GO TO 260                                       ADL55870
C                                                                       ADL55880
C     UNNECESSARY FREQUENCY OF OUTPUT                                   ADL55890
      KOP=0                                                             ADL55900
      IFLAG=7                                                           ADL55910
      GO TO 440                                                         ADL55920
C                                                                       ADL55930
260   IF (ABS(DT) .GT. U26*ABS(T)) GO TO 280                            ADL55940
C                                                                       ADL55950
C     IF TOO CLOSE TO OUTPUT POINT,EXTRAPOLATE AND RETURN               ADL55960
C                                                                       ADL55970
      DO 270 K=1,NEQN                                                   ADL55980
270     Y(K)=Y(K)+DT*YP(K)                                              ADL55990
      A=TOUT                                                            ADL56000
      CALL F(A,Y,YP)                                                    ADL56010
      NFE=NFE+1                                                         ADL56020
      GO TO 430                                                         ADL56030
C                                                                       ADL56040
C                                                                       ADL56050
C     INITIALIZE OUTPUT POINT INDICATOR                                 ADL56060
C                                                                       ADL56070
280   OUTPUT= .FALSE.                                                   ADL56080
C                                                                       ADL56090
C     TO AVOID PREMATURE UNDERFLOW IN THE ERROR TOLERANCE FUNCTION,     ADL56100
C     SCALE THE ERROR TOLERANCES                                        ADL56110
C                                                                       ADL56120
      SCALE=2./RELERR                                                   ADL56130
      AE=SCALE*ABSERR                                                   ADL56140
C                                                                       ADL56150
C                                                                       ADL56160
C***********************************************************************ADL56170
C***********************************************************************ADL56180
C     STEP BY STEP INTEGRATION                                          ADL56190
C                                                                       ADL56200
290   HFAILD= .FALSE.                                                   ADL56210
C                                                                       ADL56220
C     SET SMALLEST ALLOWABLE STEPSIZE                                   ADL56230
C                                                                       ADL56240
      HMIN=U26*ABS(T)                                                   ADL56250
C                                                                       ADL56260
C     ADJUST STEPSIZE IF NECESSARY TO HIT THE OUTPUT POINT.             ADL56270
C     LOOK AHEAD TWO STEPS TO AVOID DRASTIC CHANGES IN THE STEPSIZE AND ADL56280
C     THUS LESSEN THE IMPACT OF OUTPUT POINTS ON THE CODE.              ADL56290
C                                                                       ADL56300
      DT=TOUT-T                                                         ADL56310
      IF (ABS(DT) .GE. 2.*ABS(HH)) GO TO 310                            ADL56320
      IF (ABS(DT) .GT. ABS(HH)/0.9) GO TO 300                           ADL56330
C                                                                       ADL56340
C     THE NEXT SUCCESSFUL STEP WILL COMPLETE THE INTEGRATION TO THE     ADL56350
C     OUTPUT POINT                                                      ADL56360
C                                                                       ADL56370
      OUTPUT= .TRUE.                                                    ADL56380
      HH=DT                                                             ADL56390
      GO TO 310                                                         ADL56400
C                                                                       ADL56410
300   HH=0.5*DT                                                         ADL56420
C                                                                       ADL56430
```

```
C                                                                       ADL56440
C                                                                       ADL56450
C***********************************************************************ADL56460
C     CORE INTEGRATOR FOR TAKING A SINGLE STEP                          ADL56470
C***********************************************************************ADL56480
C     THE TOLERANCES HAVE BEEN SCALED TO AVOID PREMATURE UNDERFLOW IN   ADL56490
C     COMPUTING THE ERROR TOLERANCE FUNCTION ET.                        ADL56500
C     TO AVOID PROBLEMS WITH ZERO CROSSINGS,RELATIVE ERROR IS MEASURED  ADL56510
C     USING THE AVERAGE OF THE MAGNITUDES OF THE SOLUTION AT THE        ADL56520
C     BEGINNING AND END OF A STEP.                                      ADL56530
C     THE ERROR ESTIMATE FORMULA HAS BEEN GROUPED TO CONTROL LOSS OF    ADL56540
C     SIGNIFICANCE.                                                     ADL56550
C     TO DISTINGUISH THE VARIOUS ARGUMENTS, H IS NOT PERMITTED          ADL56560
C     TO BECOME SMALLER THAN 26 UNITS OF ROUNDOFF IN T.                 ADL56570
C     PRACTICAL LIMITS ON THE CHANGE IN THE STEPSIZE ARE ENFORCED TO    ADL56580
C     SMOOTH THE STEPSIZE SELECTION PROCESS AND TO AVOID EXCESSIVE      ADL56590
C     CHATTERING ON PROBLEMS HAVING DISCONTINUITIES.                    ADL56600
C     TO PREVENT UNNECESSARY FAILURES, THE CODE USES 9/10 THE STEPSIZE  ADL56610
C     IT ESTIMATES WILL SUCCEED.                                        ADL56620
C     AFTER A STEP FAILURE, THE STEPSIZE IS NOT ALLOWED TO INCREASE FOR ADL56630
C     THE NEXT ATTEMPTED STEP. THIS MAKES THE CODE MORE EFFICIENT ON    ADL56640
C     PROBLEMS HAVING DISCONTINUITIES AND MORE EFFECTIVE IN GENERAL     ADL56650
C     SINCE LOCAL EXTRAPOLATION IS BEING USED AND EXTRA CAUTION SEEMS   ADL56660
C     WARRANTED.                                                        ADL56670
C***********************************************************************ADL56680
C                                                                       ADL56690
C                                                                       ADL56700
C     TEST NUMBER OF DERIVATIVE FUNCTION EVALUATIONS.                   ADL56710
C     IF OKAY,TRY TO ADVANCE THE INTEGRATION FROM T TO T+H              ADL56720
C                                                                       ADL56730
  310 IF (NFE .LE. MAXNFE) GO TO 320                                    ADL56740
C                                                                       ADL56750
C     TOO MUCH WORK                                                     ADL56760
      IFLAG=4                                                           ADL56770
      KFLAG=4                                                           ADL56780
      GO TO 440                                                         ADL56790
C                                                                       ADL56800
C     ADVANCE AN APPROXIMATE SOLUTION OVER ONE STEP OF LENGTH H         ADL56810
C                                                                       ADL56820
C                                                                       ADL56830
  320   CONTINUE                                                        ADL56840
C                                                                       ADL56850
C       FEHLBERG FOURTH-FIFTH ORDER RUNGE-KUTTA METHOD                  ADL56860
C                                                                       ADL56870
C***********************************************************************ADL56880
C       FEHL INTEGRATES A SYSTEM OF NEQN FIRST ORDER                    ADL56890
C       ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM                     ADL56900
C             DY(I)/DT = F(T,Y(1),...,Y(NEQN))                          ADL56910
C       WHERE THE INITIAL VALUES Y(I) AND THE INITIAL DERIVATIVES       ADL56920
C       YP(I) ARE SPECIFIED AT THE STARTING POINT T. FEHL ADVANCES      ADL56930
C       THE SOLUTION OVER THE FIXED STEP H AND RETURNS THE              ADL56940
C       FIFTH ORDER (SIXTH ORDER ACCURATE LOCALLY) SOLUTION             ADL56950
C       APPROXIMATION AT T+H IN ARRAY F1(I).                            ADL56960
C       F1,...,F5 ARE ARRAYS OF DIMENSION NEQN WHICH ARE NEEDED FOR     ADL56970
C       INTERNAL STORAGE.                                               ADL56980
C       THE FORMULAS HAVE BEEN GROUPED TO CONTROL LOSS OF SIGNIFICANCE. ADL56990
C       FEHL SHOULD BE ENTERED WITH AN H NOT SMALLER THAN 13 UNITS OF   ADL57000
```

```
C       ROUNDOFF IN T SO THAT THE VARIOUS INDEPENDENT ARGUMENTS CAN BE  ADL57010
C       DISTINGUISHED.                                                  ADL57020
C***********************************************************************ADL57030
C                                                                       ADL57040
C                                                                       ADL57050
C                                                                       ADL57060
        CH=HH/4.                                                        ADL57070
        DO 330 K=1,NEQN                                                 ADL57080
330             F5(K)=Y(K)+CH*YP(K)                                     ADL57090
        CALL F(T+CH,F5,F1)                                              ADL57100
C                                                                       ADL57110
        CH=0.09375*HH                                                   ADL57120
C I.E. 3.*H/32.                                                         ADL57130
        DO 340 K=1,NEQN                                                 ADL57140
340     F5(K)=Y(K)+CH*(YP(K)+3.*F1(K))                                  ADL57150
C                                                                       ADL57160
        CALL F(T+0.375*HH,F5,F2)                                        ADL57170
C I.E. T+3.*H/8.                                                        ADL57180
C                                                                       ADL57190
        CH=HH/2197.                                                     ADL57200
        DO 350 K=1,NEQN                                                 ADL57210
350     F5(K)=Y(K)+CH*(1932.*YP(K)+(7296.*F2(K)-7200.*F1(K)))           ADL57220
        CALL F(T+12.*HH/13.,F5,F3)                                      ADL57230
C                                                                       ADL57240
        CH=HH/4104.                                                     ADL57250
        DO 360 K=1,NEQN                                                 ADL57260
360       F5(K)=Y(K)+CH*((8341.*YP(K)-845.*F3(K))+                      ADL57270
     1                  (29440.*F2(K)-32832.*F1(K)))                    ADL57280
        CALL F(T+HH,F5,F4)                                              ADL57290
C                                                                       ADL57300
        CH=HH/20520.                                                    ADL57310
        DO 370 K=1,NEQN                                                 ADL57320
370       F1(K)=Y(K)+CH*((-6080.*YP(K)+(9295.*F3(K)-5643.*F4(K)))+      ADL57330
     1                    (41040.*F1(K)-28352.*F2(K)))                  ADL57340
        CALL F(T+HH/2.,F1,F5)                                           ADL57350
C                                                                       ADL57360
C       COMPUTE APPROXIMATE SOLUTION AT T+HH                            ADL57370
C                                                                       ADL57380
        CH=HH/7618050.                                                  ADL57390
        DO 380 K=1,NEQN                                                 ADL57400
380    F1(K)=Y(K)+CH*((902880.*YP(K)+(3855735.*F3(K)-1371249.*F4(K)))+  ADL57410
     1         (3953664.*F2(K)+277020.*F5(K)))                          ADL57420
C                                                                       ADL57430
C       END OF FEHL SUBSECTION * * * * * * * * * * * * * * * * * * * * *ADL57440
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ADL57450
C                                                                       ADL57460
C                                                                       ADL57470
      NFE=NFE+5                                                         ADL57480
C                                                                       ADL57490
C     COMPUTE AND TEST ALLOWABLE TOLERANCES VERSUS LOCAL ERROR ESTIMATESADL57500
C     AND REMOVE SCALING OF TOLERANCES. NOTE THAT RELATIVE ERROR IS     ADL57510
C     MEASURED WITH RESPECT TO THE AVERAGE OF THE MAGNITUDES OF THE     ADL57520
C     SOLUTION AT THE BEGINNING AND END OF THE STEP.                    ADL57530
C                                                                       ADL57540
      EEOET=0.                                                          ADL57550
      DO 400 K=1,NEQN                                                   ADL57560
        ET=ABS(Y(K))+ABS(F1(K))+AE                                      ADL57570
        IF (ET .GT. 0.) GO TO 390                                       ADL57580
C                                                                       ADL57590
C       INAPPROPRIATE ERROR TOLERANCE                                   ADL57600
        IFLAG=5                                                         ADL57610
        KFLAG=5                                                         ADL57620
        GO TO 440                                                       ADL57630
C                                                                       ADL57640
390     EE=ABS((-2090.*YP(K)+(21970.*F3(K)-15048.*F4(K)))+             ADL57650
     1                        (22528.*F2(K)-27360.*F5(K)))             ADL57660
400     EEOET=AMAX1(EEOET,EE/ET)                                        ADL57670
C                                                                       ADL57680
      ESTTOL=ABS(HH)*EEOET*SCALE/752400.                                ADL57690
C                                                                       ADL57700
      IF (ESTTOL .LE. 1.) GO TO 410                                     ADL57710
C                                                                       ADL57720
C                                                                       ADL57730
C     UNSUCCESSFUL STEP                                                 ADL57740
C                       REDUCE THE STEPSIZE , TRY AGAIN                 ADL57750
C                       THE DECREASE IS LIMITED TO A FACTOR OF 1/10     ADL57760
C                                                                       ADL57770
      HFAILD= .TRUE.                                                    ADL57780
      OUTPUT= .FALSE.                                                   ADL57790
      S=0.1                                                             ADL57800
      IF (ESTTOL .LT. 59049.) S=0.9/ESTTOL**0.2                         ADL57810
      HH=S*HH                                                           ADL57820
      IF (ABS(HH) .GT. HMIN) GO TO 310                                  ADL57830
C                                                                       ADL57840
C     REQUESTED ERROR UNATTAINABLE AT SMALLEST ALLOWABLE STEPSIZE       ADL57850
      IFLAG=6                                                           ADL57860
      KFLAG=6                                                           ADL57870
      GO TO 440                                                         ADL57880
C                                                                       ADL57890
C                                                                       ADL57900
C     SUCCESSFUL STEP                                                   ADL57910
C                        STORE SOLUTION AT T+H                          ADL57920
C                        AND EVALUATE DERIVATIVES THERE                 ADL57930
C                                                                       ADL57940
410   T=T+HH                                                            ADL57950
      DO 420 K=1,NEQN                                                   ADL57960
420     Y(K)=F1(K)                                                      ADL57970
      A=T                                                               ADL57980
      CALL F(A,Y,YP)                                                    ADL57990
      NFE=NFE+1                                                         ADL58000
C                                                                       ADL58010
C                                                                       ADL58020
C                       CHOOSE NEXT STEPSIZE                            ADL58030
C                       THE INCREASE IS LIMITED TO A FACTOR OF 5        ADL58040
C                       IF STEP FAILURE HAS JUST OCCURRED, NEXT         ADL58050
C                          STEPSIZE IS NOT ALLOWED TO INCREASE          ADL58060
C                                                                       ADL58070
      S=5.                                                              ADL58080
      IF (ESTTOL .GT. 1.889568E-4) S=0.9/ESTTOL**0.2                    ADL58090
      IF (HFAILD) S=AMIN1(S,1.)                                         ADL58100
      HH=SIGN(AMAX1(S*ABS(HH),HMIN),HH)                                 ADL58110
C                                                                       ADL58120
C***********************************************************************ADL58130
C     END OF CORE INTEGRATOR                                            ADL58140
```

```
C***********************************************************************
C
C
C
C     SHOULD WE TAKE ANOTHER STEP
C
      IF (OUTPUT) GO TO 430
      IF (IFLAG .GT. 0) GO TO 290
C
C***********************************************************************
C***********************************************************************
C
C
C     INTEGRATION SUCCESSFULLY COMPLETED
C
C     ONE-STEP MODE
      IFLAG=-2
      GO TO 440
C
C     INTERVAL MODE
430   T=TOUT
      IFLAG=2
440     CONTINUE
      RETURN
      END
```